A Service Composition Model for Automatically Switched Transport Networks

Fabio Baroncelli¹, Barbara Martini¹, Luca Valcarenghi⁵, Piero Castoldi²

¹ National Laboratory of Photonic Networks - Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT)
Email: fabio.baroncelli,barbara.martini@cnit.it - Via G.Moruzzi 1, 56124, Pisa (PI), Italy
² Center of Excellence for Communication Networks Engineering - Scuola Superiore Sant’Anna
Email: valcarenghi,castoldi@sssup.it - Via G.Moruzzi 1, 56124, Pisa (PI), Italy

Abstract—Grid applications in Wide Area Networks (WANs) can benefit from the QoS-enabled dynamic connectivity offered by the Automatically Switched Transport Network (ASTN). To exploit this benefit, Grid middleware must be able to request ASTN network services with a level of abstraction that do not require the knowledge of the network implementation details. The Service Oriented Automatically Switched Transport Network (SO-ASTN) is an ASTN architectural enhancement able to export such network services by composing the ASTN connectivity services.

This work presents a Service Composition Model (SCM) and a relevant Service Composition Language (SCL) suitable for SO-ASTN. The principal novelty of the proposed SCM consists in the reuse of the Intelligent Network Service Composition paradigm in conjunction with the XML technologies used in the Web Services. The SCL, based on the ontology theory, can describe not only the composition of the ASTN connectivity but also the modality of the service process execution.

The proposed SCM and SCL are then experimentally validated on a testbed in which a Grid application requests the topology of a Virtual Private Network (VPN).

Index Terms—ASTN, Services Composition Model, Ontology Language, Grid Services

I. INTRODUCTION

Grid applications are currently moving from a dedicated Local Area Network (LAN) scenario, where all the network resources are under the customer control, to a Wide Area Network (WAN) scenario, where heterogeneous applications share the same network resources and the Quality of Service (QoS) connectivity is not guaranteed [1]. In such environment, the Grid middleware needs to be enhanced with network awareness capability to take under control the resources of network infrastructure, in particular QoS-enabled connectivity [2].

Grid middleware currently implements network awareness capability using the informative services provided by IP-based applications probes (e.g., Ping, Traceroute or Pathchar) or using executive services that allocate network resources by increasing the active application sessions (e.g., GridFTP) [3]. Within the Grid High Performance Networking Research Group (GHPN-RG) the formalization of these operating approaches is ongoing by introducing a specification of Grid Services, named Grid Network Services (GNSs), that combined with other GSs allow Grid Application to gain network awareness capability. Example of GNSs are the Network Information and Monitoring Service (NIMS) that provides up-to-date information on the Grid network status, and the Data Transport Service with Network Quality of Service (DTS-NQoS) that establishes QoS-enabled link connections among Grid Nodes [4].

The Service Oriented ASTN (SO-ASTN), presented in [5], was introduced in order to enhance the ASTN architecture with a middleware, named Service Plane (SP). By composing the connectivity services offered by the ASTN Control Plane (CP), that are named Basic Network Services (BNSs) in this work, the SP provides technology-independent network services with a level of abstraction suitable for being invoked by applications. Accordingly, applications can request network services without going through the details of the metro-core network infrastructure. In case of Grid applications, the GNSs can dynamically invoke network services to designated service nodes unburdening the Grid middleware about any technology details or actual topology of network infrastructure. The abstraction process performed by the SP must be based on a Service Composition Model (SCM) that specifies rules for the aggregation of network services in support of any application middleware. In particular, this SCM should be conceived taking into account SP architectural and performance constraints in order to be exploited by GNSs. Given the convergence of GS/GNS and Web Services (WSs) [6], the SP can adopt the WS SCM for the composition of the BNSs. Unfortunately, this solution is not feasible. In fact, WSs are discovered and invoked by applications without the need to know the entities that actually provide that WSs. On the contrary, the invocation of a connectivity service must be done directly on the specific network node that is able to satisfy that request. These arguments lead to the necessity to define a specific SCM for the network services provided by the SP.

This work presents a SCM and the relevant Service Composition Language (SCL) suitable for the SO-ASTN. The proposed SCL, based on the ontology theory, re-uses the Intelligent Network Conceptual Model (INCM) [7] defined by the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), making extensive use of XML for information representation. The strength of the proposed SCL consists in the possibility to optimize the service composition implementation by describing not only how to interconnect the BNSs, but also the execution of each BNSs (serial/parallel) and the modality of the signaling ex-
changed between BNSs (synchronous/asynchronous). Control of the execution and the signaling can be achieved because the service composition process is realized inside the SO-ASTN architecture and thus placed in a secure environment, similarly to the service composition process used in the Public Switched Telephone Network (PSTN) architecture.

The proposed SCM and SCL are applied to a use case involving Grid applications running in a WAN. This use case is related to the topology request of a Virtual Private Network (VPN) [8] issued by a Grid application and fulfilled by the SP. The use case shows the service composition process in SO-ASTN and in particular the dynamic mapping between the GNS that provide the Grid topology and the network services provided by the SO-ASTN.

The paper is organized as follows. Section II reviews the SO-ASTN architecture. Section III presents the service composition model and the relevant service composition language. In Section IV, the concept of hierarchy of service composition processes is applied to the Grid computing and a use case for providing network awareness in Grid computing is presented. Finally, in Section V the testbed set up to validate the SO-ASTN service composition model is described. Some conclusions are drawn in section VI.

II. REVIEW OF THE SO-ASTN ARCHITECTURE

The SO-ASTN [5] is a Service Oriented Architecture (SOA), depicted in Fig. 1, suitable for metro/core networks. It enhances the standard ASTN [9] by introducing a new functional plane, named Service Plane (SP), placed between the customer plane and the network infrastructure plane. ASTN and SO-ASTN have in common the following internal interfaces:

- the Network Management Interface (NMI) that supports the management functionality related to either the CP, named NMI ASTN (NMI-A), or the Transport Plane, named NMI Transport (NMI-T);
- the Reduced User to Network Interface (R-UNI) that is practically the standard UNI [10] without the security support because the SP-CP interaction is trusted;
- the Connection Control Interface (CCI) that supports the exchange of information between a node of the CP named Control Plane Node (CPN) and the controlled network element in the Transport Plane;
- the X interface that supports the exchange of information (e.g., for billing purposes) between the service provider, represented by the SP, and the transport provider, represented by the MP.

The SP satisfies customer-initiated service requests by verifying their identities and consequently arranging all actions needed for the provisioning and management of services and real-time monitoring of the relevant performance. The SP provides services to the Customer Premise Equipment (CPE) via the User to Service Interface (USI). The USI is a network technology-independent interface that exports network services hiding the service-implementation details.

As shown in Fig. 1, the proposed SP functionalities are classified in two distinct functional layers: the Centralized Service Layer (CSL) and the Distributed Service Layer (DSL).

The CSL has a centralized structure and it is responsible for all commercial aspects of service provisioning such as service marketing or billing activities related to service usage according to the agreed SLA (Service Level Agreement) contractual obligations. In particular, it manages the database containing customer information such as the stipulated SLA, current resource usage, and the corresponding access authorizations. Finally, the CSL interacts with the DSL in order to set up and release the services. The CSL is implemented by the Centralized Service Element (CSE).

The DSL is envisioned to be distributed like the ASTN CP with which it interacts. It deals with the dynamic aspect of service provisioning such as service discovery capability or actual service arrangement. The DSL is composed by a set of Distributed Service Elements (DSEs) that implement the logic for the service set-up by combining the connectivity services provided by the CP. A DSE is connected via R-UNI with one or more Edge CPNs (i.e., CPNs placed at the boundary of the network). The communication among DSEs and CSE is achieved via a dedicated signaling. The DSE is able to compose the connectivity services provided by the local CPN via R-UNI and the connectivity services provided by the remote CPNs using the DSEs as proxy servers via the SP signaling interface.

Two levels of abstraction have been distinguished for services handled by transport network: the Basic Network Service (BNS) and the Network Service (NS). Within this work a BNS represents the basic service building block of the SP that encapsulates a specific connectivity service provided by the Control Plane via R-UNI (e.g., a point to point connection with a given QoS). A Network Service (NS) is a (possibly interactive) service provided by the SP and exported at a level of abstraction suitable to be invoked by applications or application middleware (i.e., independence from the technology and topology details of the network). A NS implies the composition of one or more BNSs.

III. THE SO-ASTN SERVICE COMPOSITION MODELING

VPN Layer 1 (L1), VPN L2, and VPN L3 are three classes of network services (NS). To each class we can associate a set of mandatory primitives (e.g., VPN establishment, VPN deletion) and a set of optional primitives (e.g., VPN bandwidth on demand, VPN topology request). Since the NSs are composed by BNSs, their evolution is bound to the future enhancement of the CPN functionalities exported by the R-UNI. For this reason, the proposed SCM is open to accept future enhancements of the BNS.

The composition of BNSs realized by the SP requires (i) a SCM that specifies the set of BNSs used and the relevant interactions, (ii) a language for describing the interfaces exported by the BNSs, and (iii) a language for specifying the composition of the BNSs.

The SCM used for the SP is an adaptation of the Intelligent Network Conceptual Model (INCM) [7]. The INCM is a standardized and mature model, defined by ITU-T, suitable for connection-oriented QoS-enabled network. It was originally designed for telephone networks and it represents the first
and still complete reference model for modular service development in telecommunications. The adoption of the INCM could allow to easily re-use the telephony service infrastructure of PSTNs in the SP implementation. The INCM is based on the concept of Service Block (SB). Two types of SBs are defined in ITU-T Q.1200 Rec. series: the Functional Entities (FEs) and the Service Independent Building Blocks (SIBs). Each SB has a unique and stable interface, with one or more inputs and one or more outputs, able to exchange data with the other SBs. FE is a functional block that represents a specific network-dependent functionality (e.g., a creation of a Point-to-Point connection), a SIB is a functional block that represents a specific network-independent functionality (e.g., a Timer, a Message Counter). A generic service is obtained in the INCM by combining a set of FE and SIB. In this work a FE is represented by a BNS.

The language used in this work for describing the BNS interfaces is the Simple Object Definition Language (SODL) [11]. SODL is a simplified XML-based version of the Interface Definition Language (IDL) [12] that was the preferred language used during the last decade for describing the INCM SB interfaces. SODL allows to maintain the backward-compatibility with the INCM SBs defined in PSTN and, at the same time, to use the XML syntax. The structure of SODL is based on a few XML tags that allows the description of a generic SB interface taking into account the specific data-type (e.g., String, binary, integer, boolean) of each input/output value.

The SCL suitable for the SP is defined according to the Ontology Language principles. An Ontology Language (OL) is a formal, logical description of concepts, relations, instances, and axioms. In particular, an OL for service composition should be easily interpreted by dedicated tools for the syntax and semantic correctness verification in order to avoid incorrect use of the resources (e.g., infinite loop, lack of resource release after the end of a service). The investigated OLs are the Business Process Execution Language for Web Services (BPEL4WS) [13] and the Ontology Web Language For Services (OWL-S) [14]. Both these XML-based languages are conceived for describing the composition of WSs. Unfortunately, these languages are not effective to describe the SCM of the SP because in the WS approach, the customer directly orchestrates the services composition without dealing with service implementation details. On the contrary, in the SO-ASTN architecture the service composition is orchestrated by the network itself and the knowledge of the service implementation details allows process optimizations. For this reason, we defined the SO-ASTN Service Composition Language (SSCL) as an XML-Schema document for describing the SB composition at interface level. Even though SSCL follows an approach similar to the BPEL4WS in terms of service semantics and service-platform independence, it comprises the SBs defined in the INCM, it describes how these SBs should be connected, and it specifies the SB life cycle. SSCL is provided with the same structured-programming constructs used in BPEL4WS such as “if-then-else” and “while-do” for specifying the service control flow on the DSE that provides the network service. The SSCL novelty consists in the possibility to specify the execution process of the SB (both SIB and BNS) even in any DSE involved in the provisioning of a network service. Specifically, during the service provisioning life cycle a DSE should locally or remotely execute a set of SBs in a given order. Each SB is executed in a process and a process can execute more than one SB. A SB, called parent, can trigger the instantiation of a set of SBs called children. The SSCL allows to specify if a given SB could be instantiated in the same process of the calling SB (called “pipe”) or as a new process executed in parallel (called “fork”). In a Pipe process, the life cycle of a SB is tightly coupled to the life cycle of the parent SB thus if a DSE receives an event that implies the unsuccessful termination a BNS, the BNS children are automatically terminated as well. On the contrary, in a Fork process the life cycle of a SB is independent of the life cycle of its parent SB, thus the same event do not terminate the BNS children. Pipe and Fork processes allow to respectively implement synchronous or asynchronous interaction among
SBs, as shown in Fig. 2. This SSCL feature allows to define the composition of a service without the need to specify all the events (internal or external to the SP) that could cause the termination of a BNS thus reducing the complexity of the service composition specification.

The defined SSCL is an XML-Schema [15] file whose structure is shown in the metamodel of Fig. 3. The main XML tags are defined as follows:

- `<ServiceBlock>`: it describes a SB implemented by a DSE. It can represent a SIB/BNS, or a composition of both (a NS in this work). In the presented model the SBs are viewed by the system in terms of its observable interfaces while the internals functionalities are hidden. A `<ServiceBlock>` is a container of `<Interface>`s.
- `<Interface>`: it describes a generic interface of a SB. It is a container of input/output Data that the interface is able to exchange, and it is a container of the Action that can be invoked over itself. It adopts the same SODL XML syntax.
- `<InputData>`: it describes the type and relative constraints of a value that a specific interface is able to import.
- `<OutputData>`: it describes the type and relative constraints of a value that a specific interface is able to export.
- `<Call>`: it identifies a set of actions that can be invoked on a set of SBs and it allows to specify if they may occur in a synchronous or asynchronous mode.
- `<Action>`: it allows the invocation of a particular action with the relevant input parameters over a specified SB instance identified by its ID. In addition special Actions allow the creation of a SB.

The InputData and OutputData XML tags use the data-type defined in the SODL.

IV. GRID NETWORK SERVICES AND SO-ASTN INTEGRATION

The diagram relevant to service composition in both Grid and Network domains is shown in Fig. 4. GS/GNSs are the services of the Grid Domain. NSs and BNSs/SIBs are the two abstraction-level services present in the Network Domain that pertains to the SO-ASTN. The NSs are provided by the SP via the Grid User to Network Interface (G-UNI) that can be considered in this work as a specialization of the USI applied to Grid computing. These NSs have a level of abstraction suitable for being encapsulated by a GNS and then exported as WS using the WSDL because a network element is technology independent and it can be invoked to a single DSE without the need to know the set of DSEs actually involved in the provisioning of the network element itself. This allows Grid applications to establish and control Grid network infrastructure without having an inner knowledge of the network operations. In turn, NSs are provided by the SO-ASTN as composition of BNSs and SIBs. This allows to introduce the concept of hierarchy of service composition from the connectivity service provided by the ASTN up to the application services provided by WS. As a consequence, the NS can be reused as service building blocks in Grid Domain in order to decrease development and deployment cycle times of new Grid-enabled applications.
the Grid Nodes and, at the same time, for verifying the Service Level Agreement (SLA) stipulated with the service provider. This GNS is delivered thanks to the messages exchanged between the Grid Nodes, the DSEs, the CSE, and the CPNs.

The considered use case has the purpose of validating the proposed SCM for SO-ASTN and, in particular, it shows the hierarchical and modular implementation of GNSs starting from BNSs offered by the CP.

The network service implemented by the SP is named **VPN Topology Request** (VTR). The VTR provides the up-to-date VPN topology together with the information about the status of the virtual nodes and links involved. It is obtained as composition of a set of SIBs and BNSs as shown in Fig. 5 where the continuous line represents a synchronous call and the dotted line indicates the asynchronous call.

The BNSs implemented are the following:

- **Request Local Connection**: after receiving the identifier (ID) of a Grid VPN, it asynchronously requests the status information of all the connections established with that ID to the local CPN via R-UNI.
- **Request Remote Connection**: it is able to trigger the Request Local Connection blocks established in other DSEs of the SP.
- **Response Collector**: it is able to collect and store all the response messages, characterized by the same Grid VPN ID, coming from the other DSEs.
- **Topology Maker**: it is able to transform the XML file that contains all the information about the connectivity established in the network for a specific Grid VPN, in an XML file suitable for being interpreted by a Grid Node. This transformation was implemented thanks to the use of XSLT technology.

The SIB implemented are the following:

- **Basic Call Process (BCP)**: defined in [16] as a specialized SIB, it manages the communication with a Grid Node via USI and for each Grid service request it instantiates the relevant service session.

- **Authenticate**: defined in [17], it provides an authentication function in order to establish an authorized relationship between the service logic and a database.
- **Service Data Management**: defined in [17], it enables specific data, stored within the network, to be replaced, retrieved, incremented, or decremented according to a defined logic or by querying a lookup table stored in a database.
- **Start Session**: introduced in this work, it allows to start a process, characterized by an identifier, that can live indefinitely even after the end of this SIB. At the same time it is able to trigger a set of SBs in a predetermined order.
- **End Session**: introduced in this work, it ends a process, with a given identifier, and at the same time it is able to trigger a set of SBs, running in other processes, in a predetermined order.
- **Timer**: introduced in this work, it sends an asynchronous notification after an expiration time to a specific SB.
- **Message List**: introduced in this work, it is a queue of response message used for storing temporarily the message coming from the Grid Nodes and from the DSEs.

V. EXPERIMENTAL SETUP

The testbed used in this work is depicted in Fig. 6. The ASTN infrastructure consists of five routers, three of them configured as Edge Router, interconnected by means of Gigabit Ethernet (GE) links. The Grid network is composed by three PCs (Grid Node) running a Grid application able to generate sample traffic. The Service Plane is composed by three PCs each of them running an implementation of the DSE that control the CPN of a specific Edge Router. One PC runs an instance of CSE as well. The PCs that constitute the SP are interconnected via Ethernet links in order to support the exchange of signaling messages. Every Grid Node is connected to a router via Fast Ethernet (FE) links for data traffic.

![Testbed scheme](Image)
in section II towards the router. The G-UNI is accomplished by a socket connections supporting the exchange of XML-based documents. In fact, each NS message is expressed using XML. We simulated the R-UNI using the XML-based management interface available in most commercial routers. These interfaces are suitable for emulating UNI because they allow to create, delete, update and query the connectivity established. In addition, they allow to request information about the data traffic and the relative performance.

The Grid Nodes communicate thanks to the connectivity established by the ASTN so that the network vision of the Grid application is limited to the virtual connectivity offered by the Network Plane and presented by the SP.

The VPN Topology Request service presented in Section IV is implemented in the DSE software module. In particular, the Service Plane message set, implemented for supporting the communication between the DSE and its connected router, consists in the following messages:

- **Local Adjacency Request (LARq)** allows to query the router for obtaining the information about all the Grid connections established in that router. It is sent by the Request Local Connection FE.

- **Local Adjacency Response (LARs)** contains the list of all the connections and relative attributes established in that router for supporting the Grid network. It is sent by the Request Local Connection FE.

The used message set for the communication among DSEs consists of:

- **Remote Adjacency Request (RARq)** encapsulates a Local Adjacency Request message in order to redirect it to another DSE. It is sent by the Request Remote Connection FE.

- **Remote Adjacency Response (RARs)** encapsulates a Local Adjacency Response message in order to redirect it to another DSE. It is sent by the Request Remote Connection FE.

The sequence diagram of the message exchange among Grid applications, DSEs and router CPNs for the Grid Topology Request is shown in Fig. 7.

The Grid Application, running on Grid Node A, sends a Topology Request message to the DSE1 via GUNI interface. We suppose that the request is validated by the Authentication/Authorization block implemented in the CSE allowing DSE1 to fulfill the Grid request.

The Topology Request message is reported.

```xml
<TopologyRequest>
  <Client id=1/>
</TopologyRequest>
```

This message is an example of a simple technology-independent message supported by the G-UNI. After a request of service validation to CSE, the DSE1 sends a LARq to Router1 for obtaining information about the connections established by the Grid Nodes. At the same time, the DSE1 sends a RARq to DSE2 and to DSE3 for obtaining connectivity information from the other DSEs. Consequently DSE2 sends a LARq to Router2 and DSE3 sends a LARq to Router3. After receiving the request, each router sends a LARs to the respective DSE. Since the DSE2 and DSE3 requests were triggered by DSE1 they send a RAR to DSE1. As soon as DSE1 receives all the replies from Router1 and from its peers DSE2 and DSE3, the virtual topology of the Grid layer is calculated. Finally, a VPN Topology Response message is sent to the Grid Node A by the Topology Maker BNS. The Topology Response message is reported here below.

```xml
<Topology>
  <Node ID="100" Name="GridNodeA"/>
  <Interface ID="1">
    <Address="217.9.70.11" Type="2">
      <Port ID="1">
        <Destination NodeId="102" InterfaceId="2" PortId="1"/>
      </Port>
    </Address>
    <Performance Delay="300ms" Jitter="50" BER="7"/>
    <Bandwidth Available="10.0Mbps" Used="0"/>
    <Destination NodeId="102" InterfaceId="2" PortId="1"/>
  </Interface>
  ...
</Topology>
```

The structure of this message is very simple due to the abstraction process performed by the SP: each network node is described by a set of XML interface tags. Network topology is drawn by mutually referencing node interfaces through the attributes of the Destination tags. Each Interface is described by the link (characterized by the Port engaged jointly with the Destination tag). In turn, the link is characterized by a set of attributes (Delay, Jitter, Bit Error Ratio (BER) (expressed as $10^{-2}$), Bandwidth available, and Bandwidth used).

### VI. Conclusion

This work has presented a service composition model for the Service Plane of the SO-ASTN that allows to obtain network services starting from the basic network services offered by the CP via a technology-specific interface (R-UNI). The obtained network services are exported at a level of abstraction to be directly invoked by the network customer or by applications from a single network node thanks to a dedicated interface (USI).

In addition, a new service composition language based on the INCM paradigm and on the ontology language theory, has been presented. The novelty of this language is the possibility of describing not only the interfaces available for every SB using the SODL, but also the execution process (serial/parallel) and the modality of the signaling exchanged (synchronous/asynchronous). This language allows to improve the performance of the SP because it reduces the amount of events that a service programmer should foresee for every BNS in order to create high reactive services.

The suitability of the SO-ASTN for implementing network services for the benefit of a Grid application has been shown.
through a use case. Thanks to the SP and its SCM, network awareness in Global Grid Computing can be introduced without dealing with the complexity and the technology details of the transport network. A use case was implemented in a dedicated testbed in order to validate the SCM presented in this work.

VII. ACKNOWLEDGEMENT

This work has been partially supported by FIRB project Enabling Platforms for high-performance computational Grids oriented to scalable virtual organization (GRID.IT) (RBNE01KNFP) and by Italy-Tunisia FIRB project Software and Communication Platforms for High-Performance Collaborative Grid (RBIN043TKY).

REFERENCES