A Framework for the Establishment of Inter-Domain, On-Demand VPNs

Alexandre Matos, Fernando Matos, Paulo Simões, Edmundo Monteiro
CISUC, University of Coimbra
Pólo II, Dep. Eng. Informática
3030-290 Coimbra, Portugal
{aveloso, fmmatos, psimoes, edmundo}@dei.uc.pt

***Abstract* – Virtual Private Networks (VPNs) are a popular and cost-effective means to build wide-area corporate networks, since they provide bandwidth, privacy and security for a fraction of the cost of private networks.**

However, establishing VPNs across different domains (inter-domain VPNs) is still a cumbersome task, requiring human-based negotiation between involved carriers, manual configuration of network equipment and inefficient exploitation procedures during the whole VPN life cycle. For these reasons, inter-domain VPNs are usually contracted in a long-term basis and thus appropriate only for a limited set of usage scenarios.

In this paper we propose a framework for efficient provisioning of inter-domain VPNs, comprising a business layer for inter-carrier negotiation and orchestration, policy-based mechanisms for intra-domain resource management, and RFC 4364-based mechanisms for configuration of VPNs. This framework will allow ISPs to provide much more elastic VPN services, able to support not only traditional VPNs (contracted for long periods, for interconnection of corporate networks) but also a new class of more granular, on-demand VPNs for support of shorter-term services such as videoconference sessions or internet banking transaction.

Keywords – Inter-Domain VPNs; SOA; NGN

I. INTRODUCTION

With the current Internet ubiquity, companies acting as service providers may consider any Internet user, in any part of the world, as a potential client. Clients also have a growing desire for better services, any time they want and no matter where they are. Delivering services such as multimedia content, high-quality videoconferencing or internet banking over the internet, on the other side, imposes important security and QoS requirements.

VPNs may play an important role in this scenario, since they may provide secure channels of communication with diminished risks of interception or corruption. Associating QoS parameters with the VPNs is also a means to guarantee service requirements. When the service provider and the customer use different ISPs, VPNs will need to cross network domains. In the simplest scenario only two domains are involved (provider ISP, customer ISP) but there may also be a variable number of intermediary carriers.

However, establishing inter-domain VPNs is still an awkward task, due to the required manual configuration, the human-based negotiation between carriers and the lack of automatic mechanisms to manage the VPN life cycle. According to today's practices, establishing an inter-domain VPN requires manually exchanging negotiation and configuration parameters with all involved carriers, human-based approval of the service and explicit configuration of the network infrastructure. This situation not only increases the VPN price but also makes impractical the establishment of on-demand, carrier-provided VPNs for applications which require more granular, "immediately available" and short-term VPNs, such as banking transactions or video streaming transmissions.

These issues are not an exclusive of VPNs. Other services depending on the cooperation between multiple providers (ISPs, content providers, e-commerce portals…) also suffer similar limitations. In order to cope with these problems, there is now a general trend towards service-oriented architectures able to support service composition in Next Generation Networks (NGN) scenarios [1-7]. To leverage these paradigms into inter-domain VPN provisioning solutions is a logic step towards more competitive VPN services.

In this paper, we propose a framework for provisioning of inter-domain VPNs. This framework allows internet providers to build inter-domain VPNs in a fast and transparent way. It features a service-oriented business layer where ISPs may locate, contract and orchestrate the connection services provided by other ISPs which compose the inter-domain VPN. Policy-based mechanisms help managing resources during the service execution, and the VPN configuration on the network infrastructure is performed according to RFC 4364 [8] recommendations.

The rest of this paper is organized as follows: Section 2 discusses inter-domain BGP/MPLS VPNs. Section 3 presents related topics concerning end-to-end service provisioning. Section 4 presents the proposed framework and Section 5 discusses implementation and validation issues. Section 6 concludes the paper.

II. VIRTUAL PRIVATE NETWORKS

As already mentioned, VPNs are a cost-effective alternative to leased lines in scenarios where private channels are required or desirable. If the endpoints of the VPN are located in different domains, it becomes necessary to establish

the VPN service across two or more distinct providers (or domains). For such scenarios, BGP/MPLS VPNs conforming to [8,9] are the most popular solutions.

RFC 4026 [10] identifies the following distinct VPN routing elements: Provider Edge (PE) device, Customer Edge (CE) device, Provider device (P), and a frontier device called Autonomous System Border Router (ASBR). It is common to have many PE devices in a VPN and different alternative routes. With BGP/MPLS VPNs, the BGP protocol is used to exchange routes between PE devices, while the MPLS protocol is used to label packets. When a packet is received by the PE device, its origin is examined in order to determine the correct destination. The PE device then determines how to handle the packet, based on an address table. Each PE router keeps at least two tables: the default address table (with public and fixed routes) and the VPN Routing and Forwarding Tables (VRFs) [11]. When different domains are involved additional issues need to be addressed [12]: *topological issues* (how to interconnect the endpoints); *QoS*; *security* (common trust model and authentication mechanisms) and *management* (how to manage resources and services in VPNs that span across distinct Autonomous Systems).

These features are usually handled in a three-layer model: Service Management Layer, Network Management Layer and Network Element Management Layer (Figure 1). However, despite contributions from eTOM [13], MTOSI [14] and IPsphere [7], there are still gaps in this architecture.

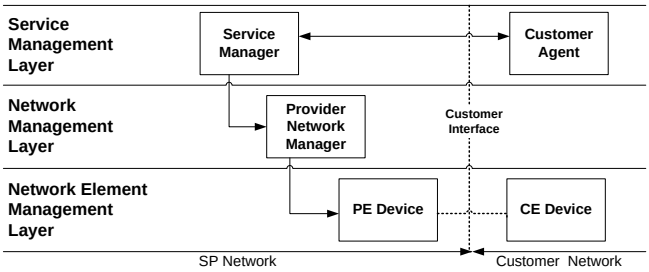

Figure 1. L3VPN Management Framework [9].

Setup and management of inter-domain VPNs is difficult due to: the dynamic addressing and topological changes; the constant need for human interference; and the heterogeneity of devices that make it complex to deal with authentication and trust relationships.

The policy-based management intrinsic to Border Gateway Protocol (BGP) guided several proposals to handle customer requirements by means of Service Level Agreements (SLAs) [15-17]. However, proprietary solutions and heterogeneity often prevent interoperability between domains.

Another constraint of current practices relates with the *static* nature of VPN services. Due to the administrative overhead involved, inter-domain VPNs are usually contracted after an exhaustive provider interaction and in a long-term basis, for static interconnection of private networks. This limits the variety of applications potentially benefiting from VPNs. Applications demanding high levels of mobility cannot use inter-domain VPNs. Other applications, by their nature, only need VPN connections for a few minutes or a couple of hours, but nowadays these dynamicity requirements are simply not met by ISPs. With more dynamic and less costly VPN provisioning, it would be possible to provide such applications with *on-demand* VPN services. In many cases users were probably willing to pay an additional fee for the benefits of such ISP-provided VPNs.

An inter-domain on-demand VPN provisioning scheme is proposed by [18]. It is based on a high level layer called Service Plane, which is used to transport signaling on the Automatically Switched Transport Network (ASTN) idealized by the authors. At Service Planes, applications have the capability to map QoS parameters into a set of directives that would then be used for service provisioning. In another proposal [19], customer authorization is needed in order to dispatch service orderings. While [18] bases these decisions on a centralized SLA database that checks authority, [19] proposes a centralized Key Management Server that dispatches authorization and service orders directly to hardware on the routers. Nevertheless, neither of these proposals integrates well into generic end-to-end service provisioning frameworks, limiting their potential interest.

III. End-to-End Service Provisioning

End-to-end service provisioning for NGN-like networks is a recurring topic, already addressed by a large number of standardization and research initiatives. It is also relevant in the inter-domain VPN scenario, since it is necessary to provide support for inter-carrier negotiation and service management. This Section reviews some of the most relevant related standardization initiatives and research projects.

The Telemanagement Forum (TMF) proposes NGOSS (Next Generation Operational Support Systems [20-21]) as a reference framework for service composition and management. NGOSS business process includes the following components: eTom (enhanced Telecom Operation Map [13]), SID (Shared Information Data and Model), TAM (Telecom Operations Map) and TNA (Technology Neutral Architecture).

eTOM identifies crucial strategies that allow business and technology to be mapped and it is considered the guide to support systems development. This guide helps customers to perceive service ordering and provisioning. However, from a multi-domain perspective, TMF focuses on MTOSI (Multi-Technology Operations System Interface [14]) which is a proposed standard for Operational Support Systems (OSS) interconnection. MTOSI is based on Service Oriented Architecture (SOA) principles and suggests Web Services and XML as tools for interconnection. Four views are identified: business, system, implementation and deployment. These views contain tasks associated to service and resources management. At business and system views, policies and agreements established as Service Level Agreements (SLAs) are input for the other views which focus on the resources where policies and agreements must be handled.

The MTOSI Implementation Lab initiative, supported by TMF, is a development group devoted to the open-source implementation of the MTOSI principles, according to the

Enterprise Service Bus (ESB) concept [22]. In the resulting framework, the Common Communication Vehicle (CCV) layer is an ESB that supports integration of providers, trough their OSSs (Figure 2). This is the medium where providers exchange service invocation. Element Management Systems (EMS) are coordinated by each MTOSI provider module, in order to translate requests for service activation.

Despite the important contribution of MTOSI, separating business logic interaction from transport features, this is still a strategy for OSS interconnection (rather than broader provider integration), overlooking for instance publication and offering of integrated services. Unlike the framework proposed in this paper, MTOSI focuses on pre-scheduled customers and providers, not dynamic, on-demand service provisioning [14].

MTOSI lacking support for service composition and NGN services is probably one of the reasons why TISPAN is also working on the harmonization of OSS systems with NGN [23], addressing areas such as utility management, service management, service platform management, connectivity management and network management.

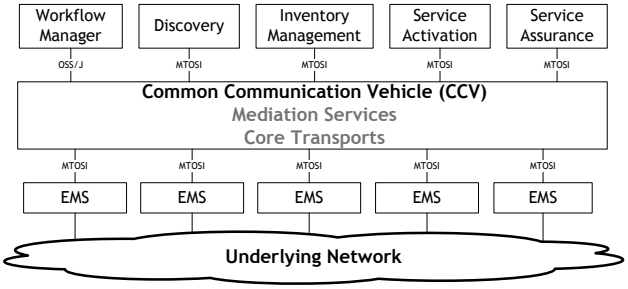

Figure 2. MTOSI general architecture (adapted from [24]).

ITIL (Information Technology Infrastructure Library [25,26]) is another standardization initiative proposing a set of recommendations for IT service management guided by business interests, including roles and SLA types for management processes. A core role in ITIL is the Service Level Manager, responsible to negotiate contracts on behalf of the customer. However, ITIL still lacks best practices for inter-domain end-to-end service provisioning.

The IPsphere Forum proposes a framework where service providers can create and expose their services without the limitations of the classical IP model [7,27]. The intent is to add a new business layer (SSS: Service Structuring Stratum) able to support all the business process necessary to locate, contract, initiate, operate, and terminate a service. Providers are interconnected with each other via Inter-Carrier Interfaces (ICI) and customers can reach IPsphere via the Customer Network Interface (CNI). In order to translate business requirements from SSS, the Signaling Network Interface (SNI) sends instructions to the lower Policy and Control Stratum (PCS), which translates high level statements in configuration instructions of its own domain. This framework is based on SOA and providers publish/find services using UDDI-enabled directory services (Figure 3). IPsphere acts like a supplement to classical network architectures, since it maps them in a framework with enhanced support for service negotiation and management.

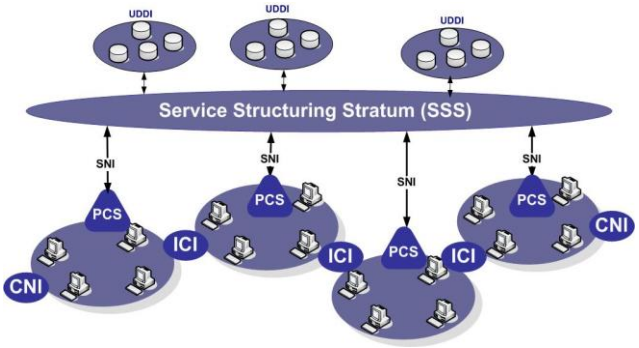

Figure 3. IPsphere framework overview.

According to the IPsphere framework, services offered to customers may be composed of multiple smaller services (*service elements*) provided by different partners (the Element Owners, EO) and assembled together by the so-called Administrative Owner (AO). The SSS layer provides an appropriate environment where business partners publish their offers and where AOs locate and contract service elements (Figure 4). Customer access to service offers is made through its provider OSS or, alternatively, third party portals (for instance a movie rental service).

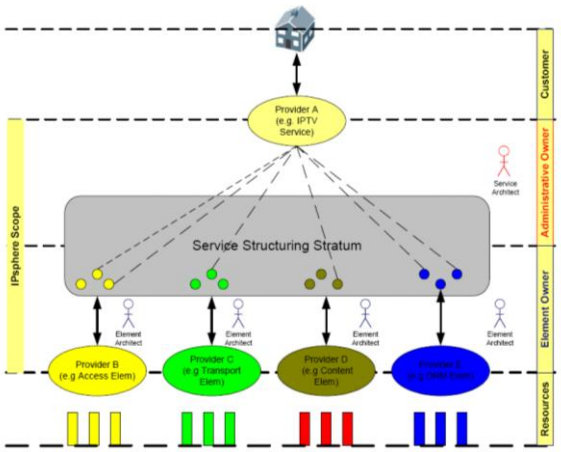

Figure 4. IPsphere flow and roles [7].

IPsphere follows NGN recommendations, including a generic transport stratum that supports multiple access networks, service monitoring/control, a mediator that enable customers to order services outside its network provider domain, service composition and other services.

The first version of the IPsphere reference architecture was recently published [7] and there are a few known ongoing implementation projects but, to the best of our knowledge, they are still far from completion. Inter-domain VPN provisioning – with the need to contract and assemble connectivity services from different providers – fits quite well into the *IPsphere* paradigm. Nevertheless, more extensive support is still desirable for issues such as trust models and service level management strategies.

Despite these standardization activities, inter-domain service provisioning still faces a number of research challenges, such as the need to support heterogeneous QoS models, the need to deal with extensive and slow negotiation

of agreements for distinct parameters, routing complexity, management of heterogeneous technological resources, distinct approaches for service management and unpredictable performance offer. From several research projects were already devoted to these issues, Tequilla and Agave were probably the most influential for current research trends.

Project Tequila [28] focused on QoS premises for an end-to-end service provisioning model. Inter-domain negotiation occurs via contracted SLAs (Service Level Agreements). However, monitoring and enforcing these agreements requires appropriate tools and mechanisms [29]. In this scenario, it is important to establish templates for documents that coordinate relationships between providers and customers and also between peer providers due to its heterogeneity. Tequilla defines, for instance, directives for template definition [30] and policy conversion and agreements [31].

Project Agave focused on the infrastructure where services are based. Traditional solutions for end-to-end IP service delivery are based on heavy communication. To lessen or even eliminate this overhead, service providers' networks are logically divided according to connectivity requirements for each service. This division is accomplished by Network Planes [32,33]. Network Planes force traffic to be classified and adequately handled according to QoS requirements. As logical partitions, network planes allow providers to compose the so called Parallel Internets (PIs), another abstraction that results from different classified flows. The Agave project, however, does not address customer interactions – in order to order or reach a service – or how PIs would be managed in order to control resources and services.

IV. Proposed Framework

What we envisage, in the future, is a business and network environment where the extra cost of establishing and managing inter-domain VPNs – when compared to the raw connectivity costs – turns out to be so low that:

- Traditional, network-based static VPNs, based on long-term contracts and dedicated to the "static" interconnection of corporate networks, will become more profitable and simpler to provide.
- And a new class of VPNs, user-based, service-driven, end-to-end, more granular and much more dynamic, will emerge and become the basis of a novel profitable market, boosting new consumer services.

In order to fulfill this vision, we propose a common framework, built on top of three keystones (Figure 5): a business layer for service assembly, policy-based automation of negotiation and resource reservation, and RFC 4364-based mechanisms for network configuration.

A. Business Layer

From a business perspective, inter-domain VPNs require a collaborative framework involving all the business partners, with clear definition of business roles and a flexible structure, able to support flexible VPN service assembly – selecting and integrating contributions from each involved player – service setup, service management and revenue share.

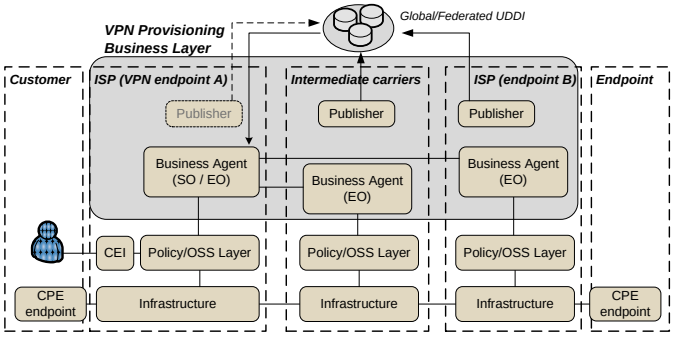

Figure 5. Global Framework Overview.

To order a VPN service, a customer uses the Customer Entry Interface (CEI). Depending on the implementation, the CEI is either a simple form offered by the ISP OSS or a more complete directory service dynamically feed by the ISP with service offers published by partner connectivity providers. In other cases the CEI is simply an API where applications such as videoconference software directly contract VPN services, sparing the customer to technical details.

Anyway, after receiving the customer request, the ISP takes the role of Service Owner (SO). Its business agent consults UDDI directories in order to locate operators (intermediate carriers and endpoint ISPs) offering connectivity services – Service Elements – potentially matching the customer request. The SO then contacts the business agents of selected partners (the Element Owners: EO), in order to get more details on price, availability and SLA conditions.

After this selection and aggregation process, the SO is able to make an offer to the customer, with the global cost and SLA of the requested VPN service. If the customer accepts it the SO contacts again the business agents of the selected EOs, in order to trigger service setup. New entities are then created to manage the lifecycle of the service (Service Instance Manager) and the service elements (Element Instance Managers). These entities are responsible for effective service creation (issuing requests to the OSS to configure network resources) and service management (monitoring, SLA assurance, service termination, etc.). As shown in Figure 6, the provider acting as SO may also act like an EO: the Service it sells is probably composed by service elements provided by other partners and service elements provided by itself (for instance last-mile connectivity).

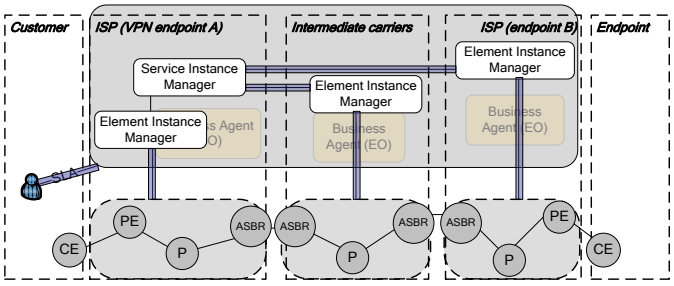

Figure 6. Service/Element Instance Managers.

To protect sensitive business data, Service Owners deal directly with each Element Owner, during service negotiation. The same happens to Instance Managers, during service

creation and management. This way each partner only knows the details about the service element he provides, not the details about the whole VPN service. The only contact between peer EOs occurs at infrastructure level (interconnection at AS border).

It is possible, nevertheless, to have chained service aggregation: an EO may sell a service element which is, itself, composed from third-party service elements. This chained aggregation, however, is always hidden.

The framework also protects business flexibility. If several EOs offer a similar service element, the SO uses internal policies to select it partners, based on technical, economic or business criteria. Besides, EOs may also differentiate their offers according to their own business policies – for instance refusing services to certain SOs or adjusting their proposal price or SLA according to the inquiring SO. This way, it is possible to range from neutral business environments (where competition is based solely on price and service quality) to closed environments, where services are sold or bought only when previous commercial agreements exist between partners.

B. Service Templates

One of the challenges associated with the proposed framework has to do with the need to correctly describe VPN services, both at customer/provider level and SO/EO level. This is accomplished using Service Specification Templates.

At SO/EO level, Element Offer templates are used to publish service offers at UDDI directories and to exchange information between the SO and the EO. Generally, UDDI directories contain sparsely filled Element Offers (with the generic information about the service), which is then further filled by the prospective SO with details about the required service and sent to the publishing EO, for enquiry. The EO then fills the Element Offer template with more detailed information (pricing, SLAs, etc.) and sends it back to the SO. Figure 7 presents a small subset of a typical Element Offer.

```
- <ElementOwner>
    <OwnerName>France Telecom</OwnerName>
  + <OwnerPostalAddress></OwnerPostalAddress>
    <OwnerEmail>ft@ft.com</OwnerEmail>
    <OwnerURL>http://www.agence.francetelecom.com/</OwnerURL>
  + <OwnerPhone></OwnerPhone>
    <ContactPersonName>Inspetor Clouseau</ContactPersonName>
  </ElementOwner>
- <ElementDescription>
    <ElementName>Transport_Connection</ElementName>
    <ElementType>Transport_Connection</ElementType>
    <PublicationDate>2005-02-05</PublicationDate>
    <ValidFor>2008-02-05</ValidFor>
  - <Monetary>
      <BasicPrice>100</BasicPrice>
      <Currency>Euros</Currency>
    </Monetary>
    <ElementDomain>ft.fr</ElementDomain>
  - <ReachableDomains>
      <Domain>bt.br</Domain>
    - <Performance>
        <worstBandwidth>10 Kb/s</worstBandwidth>
        <betterBandwidth>100 Mb/s</betterBandwidth>
        <worstDelay>20 ms</worstDelay>
        <betterDelay>0</betterDelay>
      </Performance>
      <Domain>bt.uk</Domain>
    - <Performance>
        <worstBandwidth>512 Kb/s</worstBandwidth>
        <betterBandwidth>100 Mb/s</betterBandwidth>
        <worstDelay>30 ms</worstDelay>
        <betterDelay>0</betterDelay>
      </Performance>
    </ReachableDomains>
  </ElementDescription>
```

Figure 7. Element Specification Template (fragment).

Between the customer and the SO a similar process occurs, with Service Specification Templates (Figure 8). A sparsely filled Service Offer is further completed by the customer and then sent back to the SO, which completes the template after consulting prospective EOs – for instance with global pricing, QoS information and administrative data – and sends it back to the customer, to complete the contracting process. Service specification templates include a general description section and a technical description section. This technical section, inspired by the Tequila Service Level Specification (SLS) [34], is only appropriate to applications or expert technicians. However, this is not a problem, since VPN service requests that arrive at the Customer Entry Interface are either (i) directly made by applications that hide technical details from end users; or (ii) made by specialized technicians, for instance in the case of classic corporate VPNs; or (iii) mediated by appropriate ISP customer interfaces (for instance web forms) able to hide technical complexity from the customer.

A similar process occurs with SLAs, with two distinct types of SLA: service element provisioning SLAs, agreed between the Element Owner and the Service Owner, and service provisioning SLAs, for use between the customer and the Service Owner.

```
- <ServiceOffer>
  + <ServiceOwner></ServiceOwner>
  - <ServiceDescription>
      <ServiceGlobalId>001122</ServiceGlobalId>
      <LastUpdate>2007-05-29</LastUpdate>
      <ServiceName>PortugalTelecomInterdomainVPN</ServiceName>
      <PublicationDate>2006-10-06</PublicationDate>
      <ValidFor>2007-10-06</ValidFor>
      <Category>VPN</Category>
      <SubCat>Interdomain VPN</SubCat>
    - <Scope>
        <ammountOfEndpointsSupported>10</ammountOfEndpointsSupported>
      - <multiplicity>
          <multiplicitySupported>(m,n)</multiplicitySupported>
          <multiplicitySupported>(1,n)</multiplicitySupported>
          <multiplicitySupported>(n,1)</multiplicitySupported>
        </multiplicity>
      </Scope>
    - <Monetary>
      - <BasicConfiguration>
          <minEndpoints>2</minEndpoints>
          <minTimeExpected>5 hours</minTimeExpected>
          <price>89</price>
          <currency>Euros</currency>
        </BasicConfiguration>
      - <AdditionalPrice>
          <valuePerHour>7.50</valuePerHour>
          <valuePerSite>12.45</valuePerSite>
        </AdditionalPrice>
      </Monetary>
    - <Performance>
        <worstBandwidth>512 Kb/s</worstBandwidth>
        <betterBandwidth>10 Mb/s</betterBandwidth>
        <worstDelay>30 ms</worstDelay>
        <betterDelay>0</betterDelay>
      </Performance>
    </ServiceDescription>
  </ServiceOffer>
```

Figure 8. Service Specification Template (fragment).

C. Policy/OSS Layer

This layer is responsible for the translation of business level agreements into the provider network infrastructure. From an abstract viewpoint, this translation is accomplished by means of policy-based management services that transform VPN service activation requests into router configurations according to the provider management policies and the provider network infrastructure.

From a more practical perspective, however, it should be

noted that the role of this layer corresponds to functionalities already present at classical OSS software, such as BGP/MPLS VPN service activation, monitoring, connection admission control, billing and performance and SLA management. Therefore, it is expectable that each provider implements its "Policy/OSS Layer" using the OSS software he already owns and operates. The only need for standardization is the set of interfaces with the Business Layer.

Those interfaces are not limited to service provisioning orders (VPN setup). They also provide the business layer essential information during service negotiation (admission control data, supported QoS parameters, endpoint reachability, etc.) and service management (performance and SLA management, service monitoring). Besides, the business layer also feeds, for instance, the OSS billing service.

D. Network Infra-structure, Physical Resources

This layer corresponds to the configuration of the network resources where the service is effectively executed. For inter-domain VPNs based on RFC 4364, this translates into the configuration of VPN Routing and Forwarding (VRF) entries at the autonomous system routers.

This layer has a close relationship with the policy/OSS layer. Router configuration operations and monitoring procedures are usually handled by the already mentioned OSS software, according to the interfaces available at the routers.

V. PROTOTYPE

In order to create a proof-of-concept prototype, the key components of the framework were developed and integrated in an experimental testbed.

A. Business Layer flows

Development took into account six different workflows during service life cycle:

- Service publishing. Element/Service providers publish service offers using an UDDI-based directory federation.
- Service discovery. Element/Service offers are searched at UDDI directories, according to specific criteria.
- Service creation (activation), by means of Service Owner to Element Owner negotiation.
- Service ordering by the customer.
- Service management, including monitoring of the service provisioning to guarantee its requirements.
- Service termination, after execution is completed or due to unrecoverable failures.

For the customer interface a generic API was defined for the CEI, which can be used by generic customer applications (for instance a videoconference application that hides from the customer the technical details of VPN orders). Using the same API, a web based service was developed to emulate an OSS customer care portal (the so-called B2C Portal). This service allows customers to directly contract VPN services.

In order to compose the VPN service requested by the customer, the provider who plays the role of SO uses two main sources:

- The already mentioned Element Offer template associated with each service element offer. This template describes the service technical and commercial characteristics in a standardized format, thus providing support for service element search, in a first phase, and negotiation queries, in a second phase.
- And service assembly rules. Every provider has its own set of rules used to define how to build services from service element aggregation. The framework is agnostic about these rules (each provider privately defines and implements its assembly rules), but for demonstration purposes three generic rules were implemented: cheapest solution (minimizing subcontracting costs), cheapest solution considering previous commercial agreements (the provider ignores offers from non-authorized partners), minimize delay (the provider chooses the solution that minimizes network delay).

When an SO composes a service, first it looks for the service templates to verify which elements suit service requirements. For instance, it must choose only elements whose packet loss is no more than 5% or elements that can reach certain domains. Afterwards, the SO performs a second selection based on its service assembly criteria.

After composition, the service is initiated at customer requisition. To initiate the service, the SO notifies each EO to configure the corresponding service element. The EO either rejects service element creation (which implies that the SO must restart its service assembly process) or replies with service element instantiation data. When all service elements are ready, the SO sends another message requesting service initiation. Once the service is executing, the SO is regularly informed about the status of each service element. These alerts are used by the SO to reinitiate or terminate the service, if to unrecoverable failures occur, and to react to SLA violations.

B. Policy/OSS Layer

In order to fully test the framework, a basic Policy/OSS Layer was developed, supporting the interface with the business layer and translating service provisioning requests into router configuration commands (Cisco IOS).

As already mentioned, in real usage scenarios each provider will, most likely, use OSS applications already installed to implement this intermediate layer. Therefore, it was not important to build a full-blown Policy/OSS Layer. Instead, the focus was on the validation of the interfaces defined for interaction with the business layer (service creation, monitoring and billing).

C. Network Infrastructure

The physical layer of the experimental testbed was built using the Dynamips Cisco router emulator [35] and the Dynagen front-end [36] to emulate a large number of interconnected autonomous systems populated with Cisco 7200 routers. Service establishment requests are translated, at the prototype Policy/OSS layer, into direct Cisco IOS commands which are sent to each router, in order to create, manage and terminate BGP/MPLS VPNs.

Figure 9 presents the simplest considered tested scenario, using RFC 4364 router terminology (CE: Customer Edge router, PE: Provider Edge router, P: internal Provider router, ASBR: Autonomous System Border Router). One PC was used to "host" the B2C portal, the SO, the UDDI, ISP 1 (EO, OSS layer and three emulated routers) and CE1. Another PC "hosts" ISP 2 and CE2.

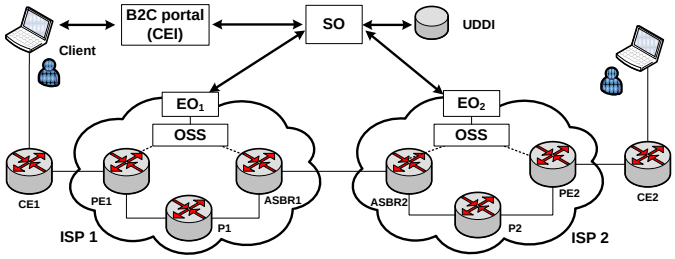

Figure 9. Validation Testbed (Basic Scenario).

D. Validation

Message exchange between the framework entities is based on Web Services, more specifically on SOAP/XML interfaces [37-38]. These technologies fit well into this kind of applications, supporting flexible and loosely coupled communication between framework modules. However, the use of SOAP/XML does not guarantee, *per se*, the desired loosely coupled architecture. It is also necessary to make sure that the devised workflows and interfaces are well designed and do not inject unexpected bottlenecks or excessive message traffic in the framework. This evaluation and optimization process, based on traffic monitoring (between business layer entities), is currently going on. Nevertheless, a few preliminary tests are presented in this paper.

Table 1 presents the network traffic associated with the establishment of one VPN, according to the scenario presented in Figure 9. Table 1 refers only to the actions necessary to establish and manage each VPN, at the business layer level, therefore excluding initial operations such as service publishing (at the UDDI service) or initial configuration of the B2C portal. Actual configuration of the routers (IOS commands issued by the OSS layer via command-line interface, and BGP configuration exchanged by routers) is also excluded from Table 1, since at that level there are no additional overheads introduced by the proposed business layer framework.

TABLE I. BUSINESS LAYER OVERHEAD

Packets (traffic octets)	*Service Composition*	*VPN establishment*	*VPN maintenance*[a]	*Service shutdown*
B2C / SO	*40 (11086)*	*7 (4331)*	-	*2 (1307)*
SO / UDDI	*46 (13950)*	-	-	-
SO / EO 1	-	*47 (17721)*	*3 (1852)*	*3 (1902)*
SO / EO 2	-	*47 (17721)*	*3 (1852)*	*3 (1902)*

a. VPN lifetime: 3 minutes

Performance measurements using the same testbed (based on PCs and heavy virtualization) showed the system could handle around four new VPNs per second, with a maximum delay for service composition around 1200 ms. Much better results are thus expectable, after prototype optimization and deployment in carrier-level servers and routers. For actual VPN establishment (including router configuration) the overhead induced by the business layer appears to be almost irrelevant, when compared with the delays resulting from routing re-configuration. More complex environments will naturally introduce additional overhead (multiple EOs available to provide the same service components; unexpected unavailability of service components, requiring negotiation; etc.), but nevertheless the overhead seems acceptable, especially when compared to the costs and performance of current practices used to contract inter-domain VPNs.

E. Implementation Remarks

Building the prototype was a mixed experience.

On one side, development of "generic-purpose" entities and interfaces was relatively straightforward. Customer Entry Interface, service publishing, service location, service negotiation and similar functionalities were not very difficult to implement. Even the definition of Service Templates adequate to inter-domain VPN scenarios, despite some effort to determine which information is necessary and relevant, was relatively simple.

On the other hand, some of the VPN specific modules were more difficult to devise. The internal logic used by the Service Owner to decide how to assemble inter-domain VPNs from different service elements, for instance, is still quite basic and not suitable for very complex environments. The effort associated with the development of the *Policy/OSS Layer* for the prototype (not intended for real use, as already discussed) showed that adding the necessary interfaces and functionalities to already existing OSS applications (the expected integration path) will be more time consuming than expected.

Since the proposed business layer is not much different from other business layers (such as the IPsphere SSS), these observations raise a relevant point: despite the importance of having generic business layer functionalities, the effort required to specifically adapt each new type of service (such as the VPN use case) to the NGN and IPsphere models should not be underrated. Hard work will still be necessary to add the intelligence, the Service Templates and the OSS support services required by each new service type.

Nevertheless, it is clear that the benefits of dynamic VPN provisioning largely compensate the required investment.

VI. CONCLUSIONS

In this paper we proposed a framework for provisioning of on-demand, inter-domain VPN services. This framework has the potential to substantially reduce administrative overheads currently associated with inter-domain VPN establishment. Furthermore, it also has the potential to boost a new generation of VPN services, more granular and dynamic.

The framework adopts NGN principles, providing a business layer where providers can offer the (partial) connectivity services which are used by the service owner (the ISP that sells the service to the customer) to assemble the complete inter-domain VPN. Bellow this business layer, the framework relies on already existing solutions to provide the connectivity service (namely existing OSS applications and

current BGP/MPLS practices). Nevertheless, the developed prototype showed that integrating these pieces – business layer, VPN-specific elements of the business layer, OSS tools, BGP/MPLS practices – is still a considerable task, that should not be underrated. This is an important observation since it probably applies to other types of services, besides inter-domain VPNs.

From the work done so far, our perspective is that NGN and IPsphere-like approaches to service provisioning are a positive step forward, despite the considerable effort involved.

Plans for future work include the assessment of the business model (from a scalability and optimization perspective) and the refinement of VPN-specific modules, such as the internal logic used by Service Owners to decide how to compose the VPN services. The current *Policy/OSS Layer* is quite basic, but since we envision tighter integration with commercial OSS applications, in the future, we have no research plans for that area. Instead, we plan to further study security and QoS issues in BGP/MPLS VPN services.

ACKNOWLEDGEMENTS

This work was partially funded by FCT (scholarship contracts SFRH/BD/16261/2004 and SFRH/BD/29103/2006) and supported by the IST FP6 OpenNet Project.

REFERENCES

[1] Y. Danfeng, Y. Fangchun, "A Universal Service Layer Management Model in NGN". Proc. of the 9th International Conference on Advanced Communication Technology, vol. 3, Feb. 2007, pp. 1721 – 1725.

[2] I. Grida et al., "Service Definition for Next Generation Networks". Proc. of the Int. Conf. on Systems and Int. Conf. on Mobile Communications and Learning Technologies (ICN/ICONS/MCL 2006), April 2006.

[3] I. Grida Ben Yahia, E. Bertin, N. Crespi, "Next/New Generation Networks Services and Management". Proc. of the International conference on Networking and Services (ICNS'06), 2006.

[4] Z. M. Yin, F. C. Yang and Y. Z. Liu, "Service management architecture and information model for next generation network with dynamic service level agreement management". Proc. of the 12th IEEE Int. Conference on Networks (ICON'04), 2004, vol. 1, Nov. 2004.

[5] E. Shervin, V.B. Lawrence, M. Malek, "The management paradigm shift: Challenges from element management to service management". Bell Labs Technical Journal, 2000. vol. 5 issue 4, pp. 3-20.

[6] N. Morita, "Introduction to NGN Functional Architecture". Proc. of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS'2006), Vancouver, April 2006.

[7] J. Alateras, "IPsphere Framework Technical Specification – Release 1". 2007, available at http://www.ipsphereforum.org/R1Spec.html

[8] E. Rosen, E. and Rekhter, Y.,, BGP/MPLS IP Virtual Private Networks (VPNs) - RFC 4364. 2006.

[9] Y. El Mghazli et al., "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management". RFC 4176, Oct. 2005.

[10] L. Andersson, T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology." RFC 4026, Mar. 2005.

[11] M. Carurig, J. De Clercq, "Virtual Private Network Services: Scenarios, Requirements and Architectural Constructs from a Standardization Perspective". IEEE Communications Magazine, Jun. 2004, pp. 116-122.

[12] M. Halstead, J. Guichard. "Requirements for Multi Autonomous System VPN Services". IETF Draft, draft-Halstead-Guichard-MAVS-Requirements-02, 2006.

[13] M. B. Kelly, "The TeleManagement Forum's Enhanced Telecom Operations Map (eTOM)". Journal of Nework and Systems Management, March 2003, vol. 11, nº 1, pp. 109–119.

[14] F. Caruso, D. Milham, S. Orobec, "Emerging industry standard for managing next generation transport networks: TMF MTOSI". Proc. of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, April 2006.

[15] S. Raghunath, K. K. Ramakrishnan, "Resource Management for Virtual Private Networks", IEEE Communications Magazine, vol. 45, issue 4, Apr. 2007, pp. 38 – 44.

[16] J.T. Park, "Management of BGP/MPLS VPN with resilient paths". Proc. of IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), April 2004.

[17] M. D'Arienzo et al. "Dynamic SLA-Based Management of Virtual Private Networks". Proceedings of Thyrrhenian International Workshop on Digital Communications (IWDC 2001), Italy, September 2001.

[18] F. Baroncelli et al., "A distributed signaling for the provisioning of on-demand VPN services in transport networks". Proc. of 10th IFIP/IEEE Int. Symp. on Integrated Network Management (IM'07), May 2007.

[19] H. Kamoda et al., "Policy conflict analysis using tableaux for on demand VPN framework". Proc. of the 6th IEEE Int. Symp. on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2005), 2005.

[20] B. M . Gupta, M. Sarkar. "Business Integration Architecture for Next Generation OSS (NGOSS)", Infosys Whitepaper, January 2006.

[21] C.R Gallen, J. S. Reeve, "Using Open Source to realise a NGOSS Proof of Concept". Proc. of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, April 2006.

[22] R. Robinson, "Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part 1", http://www-128.ibm.com/developerworks/webservices/library/ws-esbscen

[23] T. Kovacikova and P. Segec, "NGN Standards Activities in ETSI". Proc. of the 6th International Conference on Networking (ICN'07), April 2007.

[24] F. Stephen, "Multi-Technology Operations Systems Interface (MTOSI) Business Case", Telemanagement Forum, May 2006.

[25] J. Sauve et al., "An Introductory Overview and Survey of Business-Driven IT Management". Proc. of the First IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM'06), April 2006.

[26] T. Schaaf, "Frameworks for Business-driven Service Level Management: A Criteria-based Comparison of ITIL and NGOSS". Proc. of the 2nd IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM'07), May 2007.

[27] T. Nolle, "A New Business Layer For IP Networks", Business Communications Review, July 2005.

[28] D. Goderis, "D1.1: Functional Architecture Definition and Top Level Design". 2000, URL: http://www.ist-tequila.org/deliverables/D1-1.pdf.

[29] G. Cristallo and C. Jacquenet, "An Approach to Inter-domain Traffic Engineering". Proceedings of XVIII World Telecommunications Congress (WTC2002), Paris, France, September 2002.

[30] D. Goderis et al., "A service-centric IP quality of service architecture for next generation networks". Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2002). Apr. 2002.

[31] D. Agrawal, K. Lee and J. Lobo, "Policy-based management of networked computing systems". IEEE Communications Magazine, October 2005, vol. 43, issue 10, pp. 69 – 75.

[32] M. Boucadair et al., "A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets". IEEE Communications, Special Issue on QoS Control in Next Generation Networks, Sep. 2007.

[33] M. Boucadair and B. Decraene. "D1.1: Parallel Internets Framework". Available at http://www.ist-agave.org/results/D1.1-final-public.pdf.

[34] T. M. Nguyen and N. Boukhatem. "Service Level Negotiation and COPS-SLS Protocol". Proceedings of Annals of Telecommunications, vol. 59, issue 1-2, pp. 37-51.

[35] Dynamips, http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator

[36] Dynagen, http://dynagen.org/

[37] SOAP Version 1.2 Part 0: Primer: W3C Recommendation 24 June 2003. http://www.w3.org/TR/2003/REC-soap12-part0-20030624/

[38] Extensible Markup Language (XML) 1.0 (Fourth Edition): W3C Recommendation 16 August 2006, September 2006. http://www.w3.org/TR/2006/REC-xml-20060