Performance Analysis of Infrastructure Service Provision with GMPLS-Based Traffic Engineering

Slobodanka Tomic, *Student Member, IEEE,* and Admela Jukan, *Senior Member, IEEE*

***Abstract*— Dynamic sharing of the common physical network is envisioned as a key enabler for the emerging Internet technologies. This paper addresses challenges related to resource sharing in the physical layer and analyzes the performance of *infrastructure service provision* with control plane mechanisms based on Generalized Multi Protocol Label Switching (GMPLS). In our approach, the provisioning of infrastructure services is supported by two novel concepts for GMPLS traffic engineering (TE): resource visibility and inter-domain exchange. Resource visibility is a new network control plane concept, which defines the usage polices for transmission, multiplexing, and switching resources in multiple GMPLS layers. In our architecture, every network resource may exhibit different visibility to different services at different layers. The inter-domain exchange, here referred to as GMPLS Exchange Point (GXP), is the physical layer equivalent of the Internet Exchange Point (IXP). Just as how the IXP manages interconnections of autonomous systems (AS) in the Internet, the GXP manages dynamic interconnections of multiple provider domains and enables them to advertise their physical resources to other domains. We model the dynamic provisioning of infrastructure services using graph theory and deploy GMPLS traffic engineering (TE) to optimize the routing and resource yields. The results obtained demonstrate that traffic engineering with resource visibility and GXP brings significant performance benefits in resource utilization and infrastructure extensibility, especially when network providers set up LSPs as a result of collaborative and carrier-neutral traffic engineering where they share information about resource capabilities and utilization.**

***Index Terms*— GMPLS, Traffic Engineering, Multi-Layer Routing, Internet Exchange Points, Resource Management, Graph Transformation, Carrier-Neutral Provision**

I. Introduction

As a growing number of industrial, scientific and business applications are benefiting from the global Internet based on broadband optical transmission, several major networking trends emerged. First, networking communities in Grid computing [1], User-Controlled Lightpath Architecture (UCLP) [2], and the newly promoted GENI initiative [3], recommend that for the Internet to evolve it is important that network operators be able to configure multiple parallel topologies over the same physical infrastructure as the means to operate the network and to introduce new protocols and services. Second, large corporations as well as municipalities are acquiring or leasing physical resources, including fiber links, switches and routers, to build and operate their own networks, similar to the concept of Layer 1 Virtual Private Networks (L1VPN) [4]. Finally and most importantly, not only research communities but also regulatory and industry parties have realized that for network technologies to advance, the deployment of carrier-neutral open infrastructure and service provisioning are highly necessary for the growth of economies and even societies [5].

To address these developments, we introduce the notion of *infrastructure services* and analyze the performance of their dynamic provision with traffic engineering (TE) mechanisms of the Generalized Multi Protocol Label Switching (GMPLS).[1] We define an infrastructure service as an atomic infrastructure and operational entity dynamically created within the common physical infrastructure spanning multiple layers and several network domains. Every infrastructure service is running its own service-specific control plane instance including the routing function, and has direct control of the network elements allocated to it. Infrastructure services are *extensible*; in other words, while every service acts as an autonomous network, it can also attempt to dynamically acquire or release resources enabling it to run more efficiently. We expect our system to operate in a very diverse environment with multiple network providers, offering different kinds of physical resources, e.g., dark fiber, SONET-based TDM containers, wavelengths. To create and manage the infrastructure services, we introduce two novel concepts: resource visibility and inter-domain exchange.

Resource visibility is a new attribute of a physical resource which defines how the resource is used in service provision. Examples of resources include links, switches and optical routers, and examples of their visibility can be within the optical domain, TDM, IP layers, or their combinations. In our architecture, every network resource may exhibit different visibility to different services at different layers. For example, an optical router can be partially or entirely visible within a set of infrastructure services.

The GMPLS-enabled inter-domain exchange, also referred to as GMPLS Exchange Point (GXP), is similar to the Internet Exchange Point (IXP) [6], but implemented with optical switching and routing as well as with traffic aggregation (grooming) capabilities in the physical layer. Just as how the IXP manages interconnections of autonomous systems (AS) within the Internet, the GXP manages dynamic interconnections of multiple providers with GMPLS control plane. The GXP enables domains to advertise and trade their physical resources with other domains. To analyze the performance

Manuscript received April 16, 2006; revised Dec. 1, 2006.

Slobodanka Tomic is with The Telecommunications Research Center Vienna (ftw.), Vienna, Austria (e-mail: tomic@ftw.at).

Admela Jukan is with EMT-INRS, University of Quebec, Canada (e-mail: jukan@emt.inrs.ca).

Digital Object Identifier 10.1109/JSAC.2007.070603.

[1]The focus on GMPLS is mainly driven by its maturity and its intrinsic consideration of multiple layers (IP, TDM, WDM).

of infrastructure service provisioning, we develop traffic engineering methods with resource visibility and inter-domain exchange and use a graph-theoretical approach to quantify the benefits of dynamic infrastructure service availability and resource yields.

The rest of the paper is organized as follows. Section II describes the related work and states the original contribution of this paper. Section III describes the network architecture model and the concepts of resource visibility and inter-domain exchange. Section IV presents the network and service graph model, as well as the graph routing methods used to model the service provisioning. Section V describes several scenarios of infrastructure service provisioning and presents performance results. We conclude the paper and discuss open issues in Section VI.

II. Related Work and Our Contribution

Two major related directions concerning provider-provisioned VPN services have been earlier proposed for standardization within the IETF: (i) the Generalized MPLS/BGP VPN (GVPN), which reuses the proven concept of MPLS/BGP for distribution of VPN service information and MPLS tunneling [7]; (ii) the Virtual Optical Cross-Connect Service (VOXC), which reuses the concept of "virtual router" for the provisioning of VPN services [8]. While these concepts are used in the paper as the basis for infrastructure services definition, we have enriched them with the GMPLS-TE in the topology adaptation phase.

The UCLP Version 2 (UCLPv2) management tool allows for the creation of arbitrarily, user-defined topologies (called Articulated Private Networks, APNs) and is today one of the most widely used tools for experimentation in network topology configurations [2]. APNs allow many dynamically created networks to share a common network infrastructure consisting not only of links, routers, and switches, but also of instruments and end-user devices. Recently, GENI also proposed a similar approach for virtualization in the physical layer [3]. Our approach to create infrastructure services does not duplicate, but complements the UCLP approach. While UCLP is a management tool enabling network configuration at the management plane, GMPLS TE offers mechanisms of the control plane. Thus, they can co-exist and operate at different time scales.

The GXP exchange points proposed here borrow the concepts of IXPs primarily responsible of managing interconnections of autonomous systems (AS) in the current Internet [6]. Earlier in [9], the methods are proposed to determine the optimal location of peer points for data exchange among ASes. The exchange architecture based on MPLS technology called MPLS-IX was proposed in [10]. In mobile networks, the concept of resource exchange is implemented in the so-called GRX Exchange Points [11]. At every GRX, providers can dynamically negotiate resources. For example, if one mobile provider lacks capacity, it may request additional resources from its interconnected peers. As such, GRX plays a crucial role not only in supporting users' roaming but also in the proliferation of new service providers offering services without owning any infrastructure.

In the optical domain, the "distributed exchange" concept based on the optical BGP [12] was first proposed in the CA*net4 research network. The optical core acts as a reconfigurable distributed exchange point for resource reservation. Our paper [13] proposed for the first time a GMPLS-based optical exchange architecture (GMPLS-IX). In that paper, we analyzed the GMPLS-IX peering point location and the architectural difference between the GMPLS-IX architecture and the architecture with multiple UNI/NNIs. The GMPLS-based optical exchange points were later implemented and demonstrated in a testbed presented in [14].

In parallel to this research, significant efforts have been accomplished in analyzing the performance of multi-layer MPLS-based traffic engineering methods [15]. In [16], an offline GMPLS multi-layer routing approach is proposed, and it is shown that multilayer routing outperforms the single-layer approach. This is an important motivation for our analysis. In our study, we specifically show the limitations and pitfalls of multi-layer traffic engineering, when used under dynamic traffic conditions. In [17], a generic graph model for resource modeling and traffic grooming in WDM network is proposed. The novelty of our approach with respect to this one is in architecture where multiple services share physical resources based on the resource visibility attribute. Finally, we do not focus on the management plane approach for designing virtual topologies of infrastructure services, but rather evaluate the capabilities of the control plane approach. Therefore, the virtual topology design techniques, such as [18], which anticipate traffic patterns, are complementary to our study.

The present paper builds on our past research, which for the first time presented the concepts of resource visibility [13], GXPs [19] and graph transformation with multiple layers [20]. In this paper we provide the following original contributions. First, we integrate multiple concepts within a single framework, which in the past have been typically considered in isolation, including network architecture and services, network control plane and multi-layer GMPLS TE. Second, we model the GMPLS TE methods with new concepts of resource visibility and GXP without excluding any major aspects of multi-layer traffic engineering, such as forwarding adjacencies (FA) and bundled links. Third, we analyze a multi-service network operation scenario where the physical network is shared among multiple infrastructure services, including the service reserved for the best effort traffic ("background traffic"). A significant finding derived from this study is that resource visibility and multi-layer provisioning bring benefits to both customers and providers only if LSPs are setup as a result of collaborative traffic engineering policies where, unlike in the current network operations, network providers share their knowledge of resource capabilities and usage.

III. Network Architecture

Our reference network architecture is shown in Figure 1, which illustrates an infrastructure of generalized Label Switched Routers (LSR) distributed across four domains (d^1, d^2, d^3, d^4). Akin to the VPN taxonomy, we use the notion of Customer Edge (CE) and Provider Edge (PE), to distinguish among transmission, multiplexing, and generalized switching resources on the customer (user) side and on the network

provider side, respectively. Within one network domain, we denote the domain-internal resources as (P) and resources at the border between domains as (B). The resources P and B do not connect to any customer resources. Similar to L1VPN services, the infrastructure services exhibit one level of separation, i.e., within the physical layer. Every infrastructure service runs its own service control plane, which allows customers to interconnect in an arbitrary topology and also to access and control the network elements within the provider domain. In addition, the service control plane is architecturally decoupled from the network provider control plane.

To illustrate how users can utilize infrastructure services to create private networks, Figure 1 shows two infrastructure services, denoted by S_1 with four customer sites (full circles), and S_2 with three member sites (dashed). Let infrastructure service S_1 be a service managed by a regional broadcaster (RB) that wishes to link all its regional offices via an optical network and run an IP based network infrastructure over it [21]. Similarly, infrastructure service S_2 is a service used to interconnect three National Research Experimental Networks (NREN), i.e., three optical networks, to jointly run bandwidth rich experiments in astrophysics [22]. Note that for the purposes of this paper we do not distinguish between "complex" and "simple" customers/services. In other words, we not distinguish between the home user requesting a video-on-demand bandwidth pipe and a campus network operator requesting a "user-owned" network akin to UCLP. Similarly, a customer can be another provider's network, which may request a temporary expansion of capacity and infrastructure through a service like S_1 or S_2. While the customer scenarios may be different in details, they all aim at minimizing expenses of infrastructure ownership, creation, and operation, while maximizing the performance within specific time-frames. Meeting customers' demands poses a number of questions, such as: (a) how can such a framework be deployed in an existing network (b) how can the resources be chosen and interconnected, (c) how do we allow infrastructure services to perform the reconfiguration of NEs within the provider's domain?

In addressing these questions, we argue that such a framework can certainly be deployed in the existing network infrastructures that uses GMPLS TE techniques. To support the process, each service (including its end-users or service members) establishes service links, according to the service-internal traffic requirements and desired service topology. This process includes choosing and interconnecting available resources, according to the GMPLS concepts of LSP Hierarchy and LSP Regions [23] [24]. The LSP-Hierarchy reflects the multiplexing capability in the multi-layer networks with traffic aggregation capabilities (grooming). LSP-Regions are domains of a network defined within a specific layer. In other words, a particular LSP-Region includes only the interfaces which correspond to a specific switching type. In this context, every network resource in our architecture (PE, CE, P, and B) can be characterized as either an interface, or a multiplexing or switching capability in a generalized LSR. Five capability types are defined in the GMPLS Hierarchy: Packet Switching Capable (PSC), Layer-2 Switching Capable (L2SC), Time-Division Multiplex Capable (TDM), Lambda Switching Capable (LSC) and Fiber-Switching Capable (FSC). These types define five layers of granularity at which multi-layer GMPLS TE engineering can be adopted and selectively used for infrastructure service provisioning.

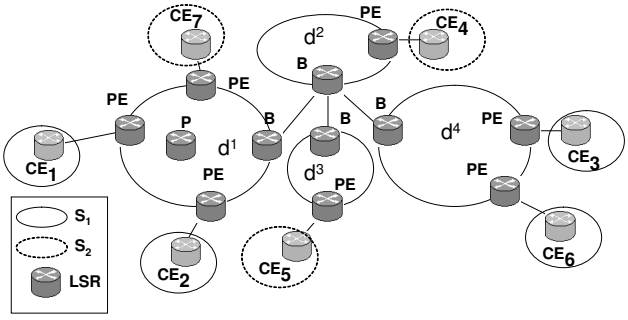

Fig. 1. The reference architecture.

A. Resource Visibility

Resource visibility is a new control plane concept, which defines the usage policies for transmission, multiplexing and switching functions in different GMPLS layers. Resource visibility is generally a parameter exposed to an infrastructure service, or more precisely, to the service control plane instance.[2] We use the term policy in context of visibility to describe the actions that can be taken by various principals (i.e., customers and providers) and require that any resource allocation request made be consistent with the policy currently active for that particular resource. The usage policies can be defined as the control plane commands that may be entered by a given principal. For example, service S_1 may be allowed to perform traffic grooming based on the resource visibility parameter.

In today's networks, TE methods are typically used to select routes based on some link parameters, such as residual bandwidth or path length. In regard to a specific resource, there is no information used for TE which indicates if and how other services use the same resource. For example, one service can acquire four switching ports, and when they are not optimally used release them temporarily for another service. Today, however, this is not possible since the related control plane information is not *shared*. We introduce the visibility parameter precisely for these scenarios, i.e., as means to share knowledge about which services can use which resources. Visibility is a rich concept which can encode the typical behavior of a service. It can also be extended by additional parameters such as trading parameter, time to use parameter, security parameter, etc. In addition, we apply this mechanism to the control and management of a resource, thus allowing the owner of the network infrastructure (network provider) to create control plane mechanisms granting the customer a certain level of control over a resource. With resource visibility, customers are given the ability to instruct NEs to perform specific tasks, such as "connect two ports of an optical switch forming a path through that switch" or "multiplex two traffic streams on NE-12".

[2]In general, a service control plane instance can be a control plane instance of another provider's domain.

It is important to notice the difference between the resource visibility and the control plane inter-operability models, e.g., peer or overlay. Interoperability models define the relationship between two control plane instances in terms of the shared topology information. Resource visibility, on the other hand, enables that customers acquire a certain level of control of network elements and to share the information about resources. In fact, both overlay and peer models can be combined with different types of visibility. In the overlay model, the resource visibility is not advertised into the service plane; service members have no access to the network elements and resource visibility is merely used within the network control plane internally. For example, if an LSP routing fails service-internally, a request for additional resources can only be *signaled* over the CE-PE control interface into the provider domain; the customer has no direct access to the resources. In the peer model, on the other hand, all resources are advertised from the provider control plane into the service control plane.

B. Inter-Domain Exchange

The infrastructure service model could be considerably improved when statically provisioned back-to-back interconnections between domains are replaced by GMPLS-enabled exchange points. Thus, we recommend that GMPLS XP (GXP) be used as the GMPLS equivalent of the Internet Exchange Points (IXPs), with the control plane instance which enables an intelligent choice of the best domain's interconnections. GXP exhibits several advantages over the architecture with multiple User Network Interfaces (UNI) and Network Network Interfaces (NNI), such as dynamic exchange of dynamic exchange of topology information, interconnection type between the domains, and reduced signaling overhead. In other words, GXP can not only support the creation of an "interconnection agreement" between the domains but it can also dynamically change the interconnection type among the domains, e.g., from 16 to 64 wavelengths. This has a significant practical relevance, since the domain's interconnects are mostly dimensioned empirically, i.e., based on measurements, and cannot dynamically adapt to the traffic dynamics.

With the increased number of domain's interconnections, e.g., hundreds and thousands, GXP can be designed to reduce the signaling overhead associated with the dynamic interconnections. Indeed, frequent updates of network state information among the domains (necessary for greater accuracy of routing) may impose a significant signaling burden. This is particularly the case for UNI/NNI architecture because of the bi-lateral nature of information exchange. Consider the case of Amsterdam Exchange Point (AMS-IX) alone, which processes 170 Gbits per second of data and interconnects around 250 Internet providers. Assuming that a control plane message uses a TLV value [24] of 41 bytes of length to periodically update all bi-lateral agreements (a conservative assumption), this would result in control plane traffic of around 1.2 Gbyte per second. With the GXP architecture, however, this can be reduced significantly, i.e., down to 20 kbyte, on the cost of processing time at the GXP control engine. Moreover, bi-lateral agreements are difficult to manage in a multi-domain network with a large number of connections. In this situation,

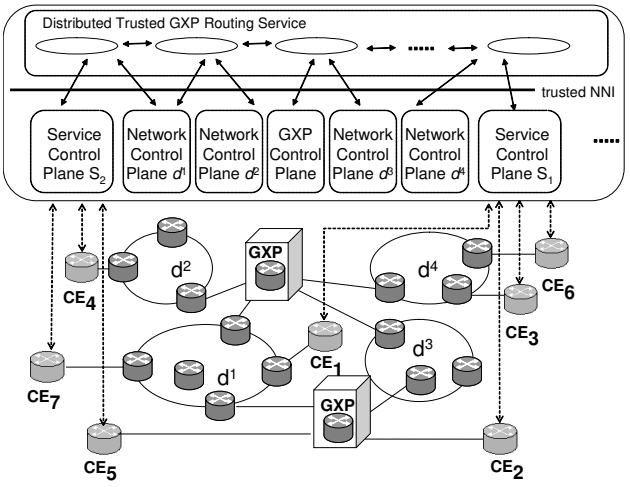

Fig. 2. Implementation of the routing overlay service with GXP-based architecture.

GXP can be used to import rules and agreements that each domain defines based on its resource management policies. The correlation of rules can now migrate to GXP, so their modification will not invoke a huge amount of information updates.

Figure 2 illustrates how the functional flexibility of the GXP can be exploited even further not only by importing policies, such as least-cost or preferred domain metrics, but also by integrating it with an GXP overlay routing service similar to that being developed within the Path Computation Element Architecture [25]. Via the GXP, routing information is imported from the service control plane and from the the control planes of the connecting domains. As a result, GXP collects the complete routing and policy information and therefore, act as a routing proxy for the connected domains and for multi-homed customers (e.g., CE-2 and CE-5). Note that Figure 2 implies that GXP also runs its own, overlaid control plane instance. In general, however, the design of an GXP control plane is an open issue that deserves a separate study. Finally, it is important to notice that GXP can also enable functions of dynamic and carrier-neutral resource trading, one of the missing functionality in today's systems [21].

IV. Network and Service Model

In this section, we present the network and service graph model as well as the GMPLS TE mechanisms for routing and resource allocation. First, we introduce the notation used in the network graph model. Second, we present the infrastructure service model featuring the *Service Visibility Graph (SVG)* and show how the *transformed SVG* is used for the service routing function. In addition, we show how SVG is updated to reflect a dynamic resource allocation. We finally present a simple TE example which illustrates the presented model. Figure 3 illustrates the relationship between the parts of the model described in detail in the subsections which follow.

A. Network Graph Model

To specify the network graph model we use the following notation.

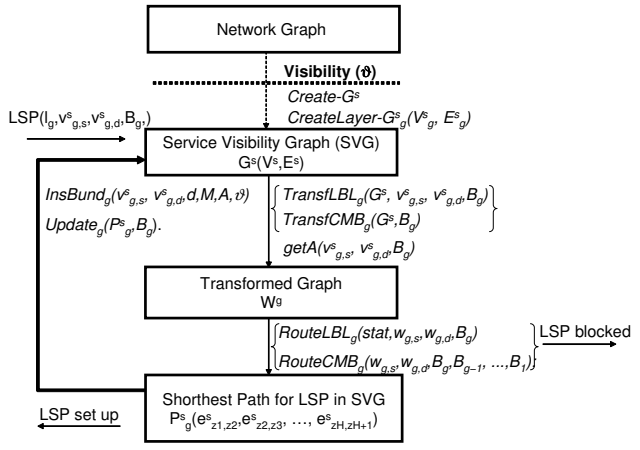

Fig. 3. Network and service graph model including the procedure calls.

Let *granularity layers* in the network be given with set $L = \{l_g\}$, $g = 1,...,|L|$. Let a lower granularity layer have a higher index, e.g., $L = \{l_1 = FSC, l_2 = LSC, l_3 = TDM\}$. Following the concepts of GMPLS, the link capacity T_{g-1} in l_{g-1} can be represented as a group of U_g containers of capacity b_g, hence $T_{g-1} = U_g \times b_g$. The values of U_g and T_g are encoded in the routing protocol extensions documents, e.g., OSPF-TE [26]. We consider U_g as a multiplexing factor between adjacent layers, i.e., l_g and l_{g-1}. Let *provider domains* in the network be given with set $D = \{d^k\}$, $k = 1,...,|D|$. Further in text k and n are used to index domains. Let *customer sites* be represented with set $C = \{c^m\}$, $m = 1,...,|C|$, and let *infrastructure services* using the network be represented with set $S = \{S_s\}$, $s = 1,...,|S|$.

Within each granularity layer $l_g \in L$ the model uses the following definitions.

Switching fabrics at the customer sites are represented with set $V^c_g = \{v^c_{g,m}\}$, $m = 1,...,|C|$. At each granularity layer a customer site $c^m \in C$ is represented with at most one vertex $v^c_{g,m} \in V^c_g$. A superscript c is used for notation of *customer* resources (nodes, network elements). The visibility attribute $\vartheta(v^c_{g,m})$ is defined as a set of services to which $v^c_{g,m}$ is visible, $\vartheta(v^c_{g,m}) \subseteq S$. We use the same principle to define the visibility for all other resources.

Let *switching fabrics of the provider domain* d^k be represented with set $V^p_{g,k} = \{v^p_{g,k,i}\}$, $i = 1,...,N_k$. A superscript p indicates that this is a *provider* resource (node, network element). The visibility of $v^p_{g,k,i}$ is denoted by $\vartheta(v^p_{g,k,i}) \subseteq S$. The switching fabrics at different layers deployed within the same resource (node) are denoted by vertex indexing. For example, $v^p_{g,k,j}$ and $v^p_{g-1,k,j}$ represent switching fabrics within the same node at layers l_g and l_{g-1} respectively.

Intra-domain links in the provider domain d^k are represented with a set of edges $E^p_{g,k} = \{e^p_{g,k,i,j} = e(v^p_{g,k,i}, v^p_{g,k,j})\}$. A superscript p indicates an intra-domain link. Every $e^p_{g,k,i,j}$ is characterized by $B(e^p_{g,k,i,j})$, corresponding to the bandwidth of the link it represents. This is expressed as a discrete number of capacity containers within layer l_g . For the same edge, the residual capacity is denoted by $R(e^p_{g,k,i,j})$, and the visibility by $\vartheta(e^p_{g,k,i,j}) \subseteq S$.

Let *access links between customer and provider sites in domain* d^k be represented with a set of edges $E^{cp}_g = \{e^{cp}_{g,k,m,j} = e(v^c_{g,m}, v^p_{g,k,j})\}$. A superscript cp indicates that the corresponding edge connects a customer with a provider. Each edge is characterized with $B(e^{cp}_{g,k,m,j})$, $R(e^{cp}_{g,k,m,j})$ and $\vartheta(e^c p_{g,k,m,j}) \subseteq S$. Similarly, *inter-domain links between domains* d^k *and* d^n are given with a set of edges $E^{pp}_g = \{e^{pp}_{g,n,k,i,j} = e(v^p_{g,n,i}, v^p_{g,k,j})\}$, characterized with $B(e^{pp}_{g,n,k,i,j})$. Residual capacity associated with this link is denoted by $R(e^{pp}_{g,n,k,i,j})$ and visibility with $\vartheta(e^{pp}_{g,n,k,i,j}) \subseteq S$. Here, a superscript pp indicates that the corresponding edge connects two provider domains.

Let *internal multiplexing and switching capacity* between the switching fabrics of the adjacent layers (l_g, l_{g-1}) of the same resource be represented with a set of edges $E^{cm}_{g,g-1} = \{e^{cm}_{g,g-1,i} = e(v^c_{g,m}, v^c_{g-1,m})\}$ for customer nodes, $E^{pm}_{g,g-1,k} = \{e^{pm}_{g,g-1,k,i} = e(v^p_{g,k,i}, v^p_{g-1,k,i})\}$ for provider nodes, respectively. The total set of all edges is denoted by $E^{um}_{g,g-1} = E^{cm}_{g,g-1} \cup E^{pm}_{g,g-1}$. Superscripts um, cm, and pm indicate that the corresponding edge represents an inter-layer (multiplexing) link. Also here, each edge is characterized with the attributes of total capacity, denoted by $B(e^{cm}_{g,g-1,m})$ and $B(e^{pm}_{g,g-1,k,j})$, respectively. The residual capacities are denoted by $R(e^{cm}_{g,g-1,m})$ and $R(e^{pm}_{g,g-1,k,j})$. The visibility is denoted by $\vartheta(e^{cm}_{g,g-1,m}) \subseteq S$ and $\vartheta(e^{pm}_{g,g-1,k,j}) \subseteq S$.

The artifacts of the GXP model are defined as follows. A set of GXP instances is denoted by $X = \{x_u\}$, $u = 1,...,|X|$. Every GXP instance x_u is an abstraction of interconnections enabled by this instance. A set of the granularity layers supported by x_u is given by L^x_u. Interconnections at the granularity layer l_g enabled by GXP x_u are given as edges $E^x_{g,u} = \{e^x_{g,u,i,j}\}$ of the graph $G^x_{g,u}(V^x_{g,u}, E^x_{g,u})$. Nodes connected to x_u are represented by a set of vertices $V^x_{g,u} = \{v^x_{g,u,i}\}$. A superscript x indicates nodes connected to a GXP and also the interconnections (edges) provided by it. The total capacity of an edge $e^x_{g,u,i,j} = e(v^x_{g,u,i}, v^x_{g,u,j})$ denoted by $B(e^x_{g,u,i,j})$, represents the interconnection bandwidth at layer l_g. The residual capacity is denoted by $R(e^x_{g,u,i,j})$, and the visibility of $e^x_{g,u,i,j}$ is denoted by $\vartheta(e^x_{g,u,i,j})$. The nodes connected by a GXP can be either provider or customer nodes, hence $V^x_{g,u} \subset V^c_g \cup_{\{d^k \in D\}} V^p_{g,k}$. We assume that at most one node per provider domain and granularity connects at each GXP.

Based on this notation, Figure 4 shows an example network graph, depicted for the network topology used in Figure 2. The graph represents a two-layer network, $L=\{l_1=LSC, l_2=TDM\}$. Inter-layer links are shown for three nodes at customer sites ($v^c_{TDM,3}$, $v^c_{TDM,4}$, $v^c_{TDM,7}$) and one node ($v^p_{TDM,2,2}$) within the provider domain d^2. Note that the GXP interconnections captured in E^x_g (GXP) are created on-demand, whereas the edges captured in E^{cp}_g (customer access links) and E^{pp}_g (back-to-back domain interconnections) are static and always present. In Figure 4, it is illustrated that GXP x_1 also interconnects customers $c_5(v^c_{LSC,5})$ and $c_2(v^c_{LSC,2})$ and not only provider domains $d^1(v^p_{LSC,1,4})$ and $d^3(v^p_{LSC,3,1})$. Through x_1, customer c_5 can connect to either or both d^1 and d^3 and exercise multi-homing. In addition, domain d^1 and d^3 can use infrastructure services to extend their corresponding infrastructures over x_1 and x_2.

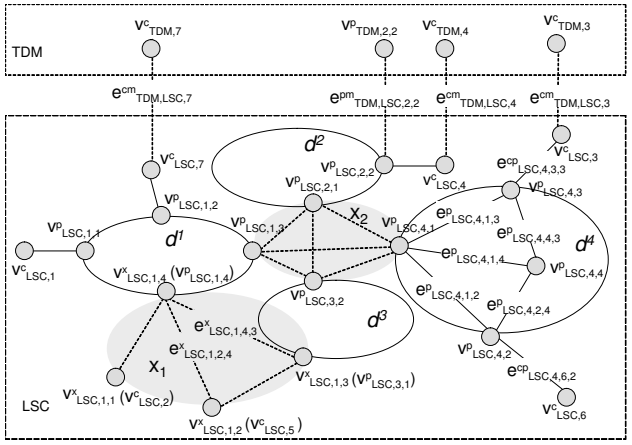

Fig. 4. Network graph model for the network shown in Figure 2 comprising four domains and two GRXs. Complete indexing scheme is illustrated in domain d^4. GRX x_1 illustrates the indexing scheme used at GRX. Inter-layer links shown are for three customer vertices and one provider vertex. Inter-domain link indexing is shown for a static, back-to-back (B-B) link between d^2 and d^3.

B. Service Graph Model

The *Service Visibility Graph (SVG)*, denoted by G^s, represents all transmission, multiplexing and switching resources visible to a service. SVG is created from the network graph based on visibility attributes, and it is dynamically updated after each TE action. We define that any arbitrary network resource r, (a vertex or an edge), is ***visible*** to S_s, if and only if $s \in \vartheta(r)$.

Service Visibility Graph (G^s) is a multi-layer graph created in two main steps, as given below. First, we create all subgraphs G^s_g for each granularity layer l_g supported by S_s, $l_g \in L^s$ (Step-1). To this aim, Step-1 invokes the procedure *CreateLayer-*$G^s_g(V^s_g, E^s_g)$. Afterwards, for each pair of adjacent subgraphs (G^s_g, G^s_{g-1}), we interconnect the corresponding vertices with edges representing multiplexing and switching capabilities. Edge $e^{sm}_{g,g-1,i} = e(v^s_{g,i}, v^s_{g-1,i})$ is referred to as service inter-layer edge. The total bandwidth $B(e^{sm}_{g,g-1,i})$, residual bandwidth $R(e^{sm}_{g,g-1,i})$ and visibility $\vartheta(e^{sm}_{g,g-1,i})$ of inter-layer edges in SVG are initialized with the corresponding values from the network graph. The edge weight $A(e^{sm}_{g,g-1,i})$ is set to a default value 1. The initial creation of the Service Visibility Graph (G^s_g) is described below.

Create-G^s

STEP-1: for each $l_g \in L^s$, invoke *CreateLayer-*$G^s_g(V^s_g, E^s_g)$
STEP-2: for each $v^s_{g,i} \in V^s_g$, if $(v^s_{g-1,i} \in V^s_{g-1})$
{ insert $e(v^s_{g,i}, v^s_{g-1,i})$ in G^s;
set $B(e^{sm}_{g,g-1,i}) = B(e^{um}_{g,g-1,i})$;
set $R(e^{sm}_{g,g-1,i}) = R(e^{um}_{g,g-1,i})$;
set $A(e^{sm}_{g,g-1,i}) = 1$;
set $\vartheta(e^{sm}_{g,g-1,i}) = \vartheta(e^{um}_{g,g-1,i})$; }

Step-1 iterates through layers and invokes the procedure *CreateLayer-*$G^s_g(V^s_g, E^s_g)$, which populates the layer specific graph (G^s_g) of SVG with all visible resources within all domains. In each layer, V^s_g is created as an ordered set of vertices representing all switching capability visible to S_s. The vertices representing visible resources of customer sites V^{sc}_g and provider domains $V^{sp}_{g,k}$, k=1, ..., $|D^s|$ within the network graph are mapped into a new set of vertices by means of a new indexing and ordering scheme, i.e., V^s_g=$\{V^{sc}_g, V^{sp}_{g,1}, ..., V^{sp}_{g,|D^s|}\}$. Correspondingly, E^s_g=$\{e^s_{g,i,j}$=$e(v^s_{g,i}, v^s_{g,j})\}$ is a set of edges E^s_g representing all visible edges from the graphs $E^{scp}_g, E^{sp}_g, E^{spp}_g, E^{sx}_g$.

The overall complexity of *Create-*G^s is determined by the number of visible links and edges and number of layers. The maximum number of vertices is bounded by $|L^s|(|C| + |N||D^s|)$, where N is the max number of nodes per domain, $|D^s|$ is the number of service domains, $|C|$ is maximal number of service members, and $|L^s|$ number of granularity layers visible within a service. The number of visible links is bounded by $|L^s|(|D^s||C| + |X^s||D^s|(|D^s|-1)/2 + |D^s|N(N-1)/2)$, where $|C||D^s|$ is a number of customer access links (CE-PE). $|X^s||D^s|(|D^s|-1)/2$ is the maximum number of interconnections at GRX, where $|X^s|$ is the number of GRX supporting the service, and $|D^s|N(N-1)/2$, is the maximum number of inter-domain edges in all domains. Hence, the complexity is bounded by $O(|L^s|(|D^s|(|C| + N) + |X^s||D^s|^2 + |D^s|N^2))$. Since the SVG configuration is performed only once, the high complexity may not impact the control plane response, as it would be the case if the SVG creation was event driven.

The dynamics of service updates becomes a more important factor during the service lifetime, when service members issue dynamic requests for setup and release of end-to-end traffic LSPs, at different granularity layers. The information dynamically maintained in SVG for each edge e^s_g includes (i) total bandwidth $B(e^s_{g,i,j})$, given as a number of containers at layer l_g, (ii) residual bandwidth, given as the number of available containers $R(e^s_{g,i,j})$, and, (iii) visibility $\vartheta(e^s_{g,i,j})$. Every container in a link can either carry traffic (dynamic state "used") or is unused (dynamic state "unused"). At the same time, SVG will sometimes need to be updated with the new dynamic TE links (FA). In order to model the dynamic changes in the SVG, we define two topology-updating procedures, $InsBund_g(v^s_{g,o}, v^s_{g,d}, M, A)$ and $Update_e(p^s_g, B_g)$. Procedure $InsBund_g(v^s_{g,o}, v^s_{g,d}, M, A)$, defined below, inserts a new dynamic FA into G^s_g, between vertices $v^s_{g,o}$ and $v^s_{g,d}$ with an edge weight A, bandwidth B, and visibility ϑ.

InsBund$_g(v^s_{g,o}, v^s_{g,d}, M, A, \vartheta_R)$

STEP-1: if $(e^s_{g,o,d} \in E^s_g)$ {goto STEP-2}
else {insert $e^s_{g,o,d} = e(v^s_{g,o}, v^s_{g,d})$ in E^s_g}
STEP-2: create $e^{scomp}_{g,o,d} = e(v^s_{g,o}, v^s_{g,d})$;
link $e^{scomp}_{g,o,d}$ to $e^s_{g,o,d}$; set $B(e^{scomp}_{g,s,d}) = M$;
set $A(e^{scomp}_{g,o,d}) = A$; set $\vartheta(e^{scomp}_{g,o,d}) = \vartheta_R$;
STEP-3: set $B(e^s_{g,o,d}) = B(e^s_{g,o,d}) + M$;
STEP-4: set $R(e^s_{g,g-1,o}) = R(e^s_{g,g-1,o}) - M$;
set $R(e^s_{g,g-1,d}) = R(e^s_{g,g-1,d}) - M$;

In Step-1, a new aggregate FA is created assuming that an edge between $v^s_{g,o}$ and $v^s_{g,d}$ does not exist. In Step-2, a component link denoted by $e^{scomp}_{g,o,d}$ with bandwidth M, weight A and visibility ϑ is created and linked (bundled) to an aggregate FA. This means that a dynamic edge in SVG is modeled as an aggregate FA which may be linked to a number of component links, where each of which has its own weight, bandwidth and visibility. An aggregate FA will then be assigned an aggregated total bandwidth, adding a new capacity M to the previous value. To establish an FA

in layer l_g the multiplexing capacity between layers l_g and l_{g-1} must be available. Therefore, in Step-4 the bandwidth of the corresponding multiplexing edges is updated (reduced by B_g) to reflect the allocation of these resources in FA. Procedure $Update_g(p_g^s, B_g)$, which is here not shown due to its simplicity, is invoked upon successful traffic routing to reflect the dynamic allocation of resources for every LSP.

The complexity of the procedure is bounded by $O(H_m U_m \log U_m)$ where H_m is the maximum number of hops in the path and U_m is the maximal number of containers in one link. (In Section IV-C we show how the path is found.)

In order to develop different routing strategies, we can use specific procedures $getA(v_{g,o}^s, v_{g,d}^s)$ to model various dynamic weighting schemes. Procedure $getA(v_{g,o}^s, v_{g,d}^s)$ calculates the weight A of the edge $e_{g,o,d}^s$ in SVG between the vertices $v_{g,o}^s$ and $v_{g,d}^s$ when it is used for routing of an LSP request for B_g containers of l_g granularity. This procedure returns a dynamically calculated weight (here a weight of a min-cost component link). It is important to note that only an aggregate FA is used for routing and therefore it must be characterized by a unique composite weight. Since parallel (component) FA links can have different weights (as this weight can reflect the properties of the underlying LSP, such as the number of hops), calculating the composite weight is challenging. In this study, the weight of the aggregated FA link is set to the minimum weight of all component FA links that have enough capacity to support a traffic request. The complexity of $getA_g(v_{g,o}^s, v_{g,d}^s)$, shown below, is bounded by the complexity of a binary sort $O(U_m \log U_m)$ where U_m is the maximum number of component links in one FA.

$\underline{getA_g(v_{g,o}^s, v_{g,d}^s, B_g)}$

STEP-1: find all $e_{g,o,d}^{scomp*} \subseteq e_{g,o,d}^{scomp}$, where $e_{g,o,d}^{scomp*}$ linked to $e_{g,o,d}^s$ and $R(e_{g,o,d}^{scomp*}) > B_g$.

STEP-2: find $A = \min_{\{e_{g,o,d}^{scomp*}\}}\{A(e_{g,o,d}^{scomp*})\}$;

STEP-3: return A.

C. Service Traffic Routing and TE Link Setup

We denote a traffic request by $LSP(l_g, v_{g,o}^s, v_{g,d}^s, B_g)$, where l_g is the granularity layer, B_g is the requested capacity, and (service s members) $v_{g,o}^s$ and $v_{g,d}^s$ are the LSP's origin and destination, respectively.[3] Every LSP request results in allocation of resources and potentially in a setup of new dynamic service links. To deal with scenarios of multi-layer resource allocation, we define the multiplexing factor U_g as the maximum number of l_g containers within one l_{g-1} container, i.e., in the adjacent layer. In this way, a request for B_g containers at l_g can be translated into a request for $int(B_g/U_g)$ containers at the layer l_{g-1}. For example, one wavelength (LSC layer) is an optical channel carrying four STM64 containers; U_{TDM}=4 and a request for five STM64 maps into a request for two wavelengths.

For the purposes of routing, an SVG has to be transformed into a subgraph W_g with a set of vertices and edges denoted by $\{w_{g,i}\}$ and $\{e_{g,i,j}^w = e(w_{g,i}, w_{g,j})\}$, respectively. Each edge in W_g is associated with a weight $A(e_{g,i,j}^w)$, calculated in the procedure $getA(v_{g,i}^s, v_{g,j}^s)$ described previously. Every request $LSP(l_g, v_{g,o}^s, v_{g,d}^s, B_g)$ in SVG maps to a request $LSP(w_{g,o}, w_{g,o}, B_g)$ in W_g. The shortest path P_g found in W_g is expressed as an ordered set of edges $P_g(e_{g,z^1,z^2}^w, e_{g,z^2,z^3}^w, ..., e_{g,z^H,z(H+1)}^w)$, where H is the hop count of P_g. For every P_g, we denote the corresponding path in SVG with $P_g^s(e_{g,z^1,z^2}^s, e_{g,z^2,z^3}^s, ..., e_{g,z^H,z(H+1)}^s)$, which is then used to dynamically update SVG (i.e., procedure $Update_g(p_g^s, B_g)$).

In this setting, there are two basic ways to approach routing, i.e., layer-by-layer or layers-combined. The layer-by-layer approach considerers separately each layer when routing a traffic request. The higher granular layer l_{g-1} acts as a server satisfying a request for the topology extension coming from the layer below, i.e., l_g. The weighting schemes of these two layers can be completely separated. The layers-combined approach, on the other hand, considerers resources of different layers at the same time.

1) Layer-by-Layer (LBL) Routing: LBL routing uses layer specific subgraphs of the corresponding SVG to execute the routing at each layer separately. The transformed graph, referred to as W_g, is created by first replicating all vertices from the service graph at l_g layer. After that, only the edges with sufficient bandwidth are included in the transformed graph. Every other step is the same as in $Create\text{-}G^s$, with only difference that also inter-layer edges between SVG subgraphs G_g^s and G_{g-1}^s are included. They indicate that a new higher granular LSP may be initiated within a higher layer, when necessary. The transformation of the SVG procedure is denoted by $TransfLBL_h(G^s)$. The complexity of this procedure for W_g creation is determined with the complexity of the graph search and is therefore comparable to the complexity of the procedure $Create\text{-}G^s$.

Procedure $RouteLBL_g(w_{g,o}(v_{g,o}^{sc}), w_{g,d}(v_{g,d}^{sc}), B_g)$ uses the SVG transformation to accommodate a request for $LSP(l_g, v_{g,o}^{sc}, v_{g,d}^{sc}, B_g)$. As a result a path is found which may be a composite of successive LSPs at different layers. During the procedure shown below, it is important to note that the layer of the original request is captured with the variable $stat$.

$\underline{RouteLBL_g(stat, w_{g,o}, w_{g,d}, B_g)}$:

STEP-1: if ($stat$ = "") {set $stat$ = "l_g"}

STEP-2: find the shortest path $P_g(e(w_{g,o}, w_{g,z^2}), ..., e(w_{g,z^H}, w_{g,d}))$, if P_g exists {goto STEP-5} else {goto STEP-3}

STEP-3: set $h = g - 1$; if ($e_{g,g-1,o}^w$, $e_{g,g-1,d}^w \in W_g$) {goto STEP-4} else {set $stat$ = "$blocked$"; goto STEP-9};

STEP-4: $TransfLBL_h(G^s)$; set B_h=$int(B_g/U_g)$; $RouteLBL_h(stat, w_{h,o}, w_{h,d}, B_h)$; goto STEP-9;

STEP-5: $Update_g(P_g^s, B_g)$;

STEP-6: if ($stat \neq$ "l_g") {goto STEP-7} else {set $stat$ = "$sucess$"; goto STEP-9}

STEP-7: set $h = g + 1$; insert $e(w_{h,o}, w_{h,d})$ in W_h; set A($e(w_{h,o}, w_{h,d})$) = $(H - 1)$; set $\vartheta_r = \cap_{(e_{h,i}^w \in P_h^s)} \vartheta(e_{h,i}^w)$; $InsBund_h(v_{h,o}^s, v_{h,d}^s, B_h, (H-1), \vartheta_r)$;

STEP-8: $RouteLBL_h(stat, w_{h,o}, w_{h,d}, B_h)$;

Here, if the routing in l_g fails, an *extension* request for an LSP in layer l_{g-1} is generated between the source and

[3] An additional parameter τ, the duration of the requested LSP may also be specified and used for routing. In this study, we do not consider the duration τ as a routing parameter, as it deserves a separate study.

destination node of the original LSP request, using the multiplexing factor U_g to calculate bandwidth requested in l_{g-1}, i.e., $B_{g-1} = int(B_g/U_g)$. Further (Step-4), the graph W_{g-1} is created in the procedure $TransLBL_{g-1}()$ and the procedure $RouteLBL_{g-1}()$ enters the next iteration (Step-1) with the variable $stat$ already initialized. In the case of a route search failure, the next lower layer is entered or if such does not exist, the request is blocked (Step-3). In the case of successful routing, the procedure differentiate between the original LSP layer and any other layer (Step-6) based on the value of $stat$. In any layer l_g other than the original LSP layer Step-7 is executed. Here, the transformed graph of the next higher layer W_{g+1} is updated with a new edge representing a dynamically created FA. Dynamic FA link is assigned a weight equal to $(H-1)$, where H is the number of hops of the underlying LSP at layer l_{g-1}, as proposed in [27]. SVG in the higher layer G^s_{g+1} must also be updated in the procedure $InsBund_g$; here, a new edge is bundled to an existing edge and updated with the allocated LSP capacity at layer l_g, i.e., B_h. The visibility of a new link is calculated as the common subset of visibility sets at all edges in P^s_g. The routing procedure is then restarted at the higher granularity layer. Eventually, the initial request will be completed. The procedure can iterate through all layers visible within the service and the maximum number of iterations is $|L^s|$.

The complexity of this procedure is determined by the complexity of the shortest path search, which may be executed maximally $2|L^s|$ times, hence the complexity is bounded with $O(|D^s||C| + |X^s||D^s|^2 + |D^s|N^2))$, if the Dijkstra shortest path algorithm is used. Procedures $TransfLBL_h(G^s)$, and $Update_g(P^s_g, B_g)$ are executed $|L^s|$ times at most. Thus the overall complexity is bounded by $O(|L^s|(H_m U_m \log U_m + |D^s|(|C| + N) + |X^s||D^s|^2 + |D^s|N^2))$, where N is the max number of nodes in all domains, $|D^s|$ is the number of service domains, $|C|$ is maximal number of service members, $|L^s|$ number of granularity layers visible within service, U_m maximal number of component links in one aggregated FA, H_m is the maximal number of path hops, and $|X^s|$ number of GRXs supporting the service.

2) Combined Routing: The graph transformation procedure for CMB routing $TransfCMB_g(G^s, B_g)$ is more complex than $TransfLBL_h(G^s)$ as it includes all visible resources at all layers.

$TransfCMB_g(G^s, B_g)$:

STEP-1: for each $v^s_{g,i} \in V^s_g$, create $w_{g,i}(v^s_{g,i})$ in W_g;
STEP-2: for each $e^s_{g,i,j} = e(v^s_{g,i}, v^s_{g,j}) \in E^s_g$
if $(B(e^s_{g,i,j}) > B_g)$
{insert $e^w_{g,i,j} = e(w_{g,i}(v^s_{g,i}), w_{g,j}(v^s_{g,j}))$ in W_g;
set $a(e^w_{g,i,j}) = A(e^s_{g,i,j})$}
STEP-3: for each $l_h \in L^s$, start with l_{g-1} end with l_1,
if $(B(e(v^s_{h+1,s}, v^s_{h,s})) > B_{h+1}$ and
$B(e(v^s_{h+1,d}, v^s_{h,d})) > B_{h+1})$
{set $B_h = int(B_{h+1}/U_g)$; goto STEP-4}
else {goto STEP-8}
STEP-4: for each $e^s_{h,i,j} = e(v^s_{h,i}, v^s_{h,j}) \in E^s_h$
if $(w_{g,i} \in W_g)$ {set $w_a = w_{g,i}$; goto STEP-5}
else {insert $w_{h,i}(v^s_{h,i})$ in W_g;
set $w_a = w_{g,i}$; goto STEP-5}
STEP-5: if $(w_{g,j} \in W_g)$ {set $w_b = w_{g,j}$; goto STEP-6}
else {insert $w_{h,j}(v^s_{h,j})$ in W_g;
set $w_b = w_{h,j}$; goto STEP-6}
STEP-6: if $(e(w_a, w_b) \in W_g)$ {goto STEP-3}
else {goto STEP-7}
STEP-7: if ($B(e^s_{h,i,j}) \times U_h \times ... \times U_g > B_g$)
{insert $e(w_a, w_b)$ in W_g;
set $type(e(w_a, w_b))$ = "$transit_h$;
set $a(e(w_a, w_b))$ = 1; goto STEP-4;}

In Step-1 and Step-2, similarly to LBL, we include in W_g the vertices and edges from SVG layer l_g that have sufficient bandwidth. In Step-3, the inter-layer edges between l_g and l_{g-1} in SVG are also checked for sufficient capacity. For each edge with sufficient capacity the vertex from layer l_{g-1} is inserted in the transformed graph W_g (Step-4). Only those vertices from l_{g-1} are inserted that do not already have a corresponding representation in W_g. In addition, the edges with sufficient bandwidth in SVG layer l_{g-1}, that are not already represented in W_g are also inserted and marked as $transit_{g-1}$. Transit edges represent resources from all existing layers in the same graph W_g. The weight of each transit link is set to a default value 1. Note that the sufficient bandwidth is calculated based on the multiplexing factors at different layers. The insertion of vertices and edges is repeated for each subsequent layer, starting with l_1.

The complexity of this procedure is bounded by the complexity of the graph search, i.e., $O(|L^s|(|D^s|(|C| + N) + |X^s||D^s|^2 + |D^s|N^2))$, where N is the max number of nodes in all domains, $|D^s|$ is the number of service domains, $|C|$ is maximal number of service customers, and $|L^s|$ number of granularity layers visible within a service.

The combined routing is described as procedure $RouteCMB_g(w_{g,s}, w_{g,d}, B_g, B_{g-1}, ..., B_1)$, where $w_{g,s}$ and $w_{g,d}$ are the source and destination of the request, and B_g is the LSP's requested bandwidth. The values of $B_{g-1}, ..., B_1$ are calculated in $TransfCMB_g(G^s, B_g)$.

$RouteCMB_g(w_{g,o}, w_{g,d}, B_g, B_{g-1}, ..., B_1)$:

STEP-1: find the shortest path P_g given with
$P_g(e(w_{g,o}, w_{g,z^2}), ..., e(w_{g,z^H}, w_{g,d}))$,
if P_g exists {$stat$ = "$success$"; goto STEP-2}
else {$stat$ = "$blocked$"; goto STEP-4}
STEP-2: for each $l_h \in L^s$, start with l_1 end with l_{g-1}
{find concatenations of $transit_h$ links $Q_{r,h}$ in P_g;
for each found $Q_{r,h}$, with the length $r < H$, given with
$Q_{r,h} = (w_{h+1,l^a}, w_{h,l^a}, ..., w_{h,l^{a+r}}, w_{h,l^b}, w_{h+1,l^b})$
{$Update_h(Q^s_{r,h}, B_h)$;
substitute $Q_{r,h}$ with $e(w_{h+1,l^a}, w_{h+1,l^b})$ in P_g;
set $type(w_{h+1,l^a}, w_{h+1,l^b})$="$transit_{h+1}$";
set $\vartheta_r = \cap_{(e^w_{h,i} \in Q^s_{r,h})} \vartheta(e^w_{h,i})$;
$InsBand_{h+1}(v^s_{h+1,l^a}, v^s_{h+1,l^b}, B_{h+1}, H^r\text{-}1, \vartheta_r)$; } }
STEP-3: $Update_g(G^s, P_g, B_g)$;

In Step-1, if the shortest path P_g is not found, the request must be blocked, since the available resources at all layers do not form a connected graph. If a path is found, SVG and the transformed graph are updated in each layer l_h (Step-2) starting from the highest granularity layer l_1, and ending with l_{g-1}. The path P_g is searched for concatenations of transit links at different layers (Step-3). For each concatenation $Q_{r,h}$ in P_g of transit edges of type "$transit_h$", we first find the corresponding path $Q^s_{r,h}$ in G^s and invoke an SVG update ($Update_h$). In this way, the link resources in $Q^s_{r,h}$ are allocated in G^s_h. Note that the bandwidth allocated, i.e.,

B_h, is already calculated in the $TansfCMB$ procedure by taking into account the multiplexing factors of all layers. A concatenation of transit edges $Q_{r,h}$ will be substituted in P_g with a single edge between the ingress and egress nodes of $Q_{r,h}$, and type of this single edge will be set to "$transit_{h+1}$". After that, the procedure $InsBundle_{h+1}$ is invoked in order to insert a new dynamic link in the service graph G^s_{h+1}. The weight of the inserted link is set $(H^r - 1)$, where H^r is the number of hops in $Q_{r,h}$ [27]. The bandwidth is set to B_h. The visibility of a new link is calculated as the common subset of visibility sets at all edges in $Q^s_{r,h}$. In the loop of this process, all concatenations of transit links at all layers will be gradually substituted until the path P_g consists only of links in l_g layer. In Step-3, the procedure $Update_g()$ updates the resource information in G^s_g which corresponds to the requested traffic LSP.

The complexity of the routing procedure is determined by the complexity of the shortest path routing, as well as by the search for concatenations of transit edges ($O(H_m \log H_m)$) and the update procedure ($O(U_m \log U_m)$). Thus the overall complexity is bounded by $O(U_m \log U_m + H_m \log H_m + |D^s|(|C| + N) + |X^s||D^s|^2 + |D^s|N^2))$, where N is the max number of nodes in all domains, $|D^s|$ is the number of service domains, $|C|$ is maximal number of service members, $|L^s|$ number of granularity layers visible within service, H_m is the maximal number of path hops (in one layer), and $|X^s|$ number of GRXs used. Optimization and reduction of complexity may include the consideration of shortest path algorithms faster than Dijkstra, and a reduction of the number considered layers and domains.

D. TE Link Release

The infrastructure services *extensibility* does not only mean that a service is able to *extend connectivity*, but also to *reduce connectivity*. The latter is equivalent to releasing of the TE links and de-bundling them from a service. Every service can independently decide whether to release dynamic links upon the release of the LSP traffic, or to reuse it for new requests in course of future TE actions. Releasing TE links may be invoked from within the service control plane or the provider control plane. We assume that a background control plane process is checking the usage of all TE links periodically, and based on the result of this action, a release of resources can be timely requested.

We define three TE link release strategies: *never release* (NR), *release when idle* (REL) and *scheduled release* (SREL), which may be applied to any dynamically established TE link. A *never release* strategy keeps a dynamically created TE link configured for the whole service duration. A *release when idle* releases a TE link as soon as its utilization falls below a certain threshold, for example if it is not used during a specific measurement interval. Therefore, the *release when idle* strategy can help to better share the underlying transmission and switching resources. The third strategy, i.e., *scheduled release* assumes that a resource scheduler can mark unused resources as "not to be used for further requests". In this case, the unused TE link is avoided for future requests, and eventually, after a scheduled resource availability check, is labeled as unused and released. This strategy can support a graceful preemption. Release procedures are not formally described here, as they are just an inverse of the procedures $Update_g$ and $InsBund_g$.

E. TE Actions in Two-layer Scenarios

We illustrate the methods previously presented with an example in a two-layer network with LSC and TDM grooming and switching capability at the customer equipment and in the provider domain, i.e., $L = \{l^1 = LSC, l^2 = TDM\}$. For simplicity and without loss of generality, we consider a single service S_1, provisioned within a provider domain, with LSP TDM traffic requests. In this example, we define two visibility scenarios, which we refer to as *reduced visibility* and *extended visibility*. In the *reduced visibility* scenario, the TDM-layer resources in the provider domains are not visible, i.e., cannot be used for the service traffic. In the *extended visibility* scenario the TDM grooming resources in the provider domain are made visible to the service. In both cases, we assume a full visibility of LSC resources.

Figure 5 shows an SVG created for a simple network topology, for reduced and extended visibility. The network topology includes three service members $(v^s_{LSC,1}, v^s_{LSC,2}, v^s_{LSC,3})$ and three LSRs in the provider domain $(v^s_{LSC,4}, v^s_{LSC,5}, v^s_{LSC,6})$. SVG is composed of two subgraphs G^s_{TDM} and G^s_{LSC} which are interconnected with inter-layer edges $e^{sm}_{TDM,LSC}$, depicted as dashed. For both visibility cases G^s_{LSC} includes vertices which represent the LSC switching capability at the customer and provider sites.

In the *reduced* visibility case, G^s_{TDM} includes only vertices representing TDM switching capability at the customer sites G^s_{TDM} and also shows three dynamic links $e^s_{TDM,1,2}$, $e^s_{TDM,3,2}$ and $e^s_{TDM,1,3}$. The edge $e^s_{TDM,1,2}$ may be for example an aggregated FA composed of FA_1 created over LSC-LSP $(v^s_{LSC,1}, v^s_{LSC,4}, v^s_{LSC,5}, v^s_{LSC,2})$ and FA_2 created over LSC-LSP $(v^s_{LSC,1}, v^s_{LSC,4}, v^s_{LSC,6}, v^s_{LSC,5}, v^s_{LSC,2})$.

In the *extended* visibility scenario, TDM switching capable nodes are also visible in G^s_{TDM} and therefore also included in G^s_{TDM}. The set E^s_{TDM} may include up to 30, , $i.e., 6 \times (6-1)$, dynamically created FAs. In Figure 5, E^s_{TDM} includes 10 FAs.

Figure 6 illustrates the difference between the transformed graphs for LBL and CMB methods, under the assumption of reduced visibility. The assumption is that edges in G^s_{TDM} do not have enough bandwidth and are not included in W_{TDM}. In W_{TDM}, which is created with the LBL method, only the inter-layer links (dashed) are included, which indicates a possibility of graph connectivity extension in the LSC layer. In W_{TDM} created with CMB, on the other hand, the transit links from G^s_{LSC} are included and shown as dashed. These edges represent a path that connects the otherwise disconnected TDM topology. In this case, since LBL routing between $w^s_{TDM,1}$ and $w^s_{TDM,2}$ does not result in a routing path, an graph connectivity extension in W_{LSC} is started. In this case, an LSC-LSP is set up, resulting in a new FA between $v^s_{TDM,1}$ and $v^s_{TDM,2}$. This FA will be then used to accommodate the traffic TDM-LSP traffic request. The CMB routing finds the path over the transit links which are then used to set up the corresponding LSC-LSP and a dynamic FA. The dynamic FA is used for a TDM LSP request.

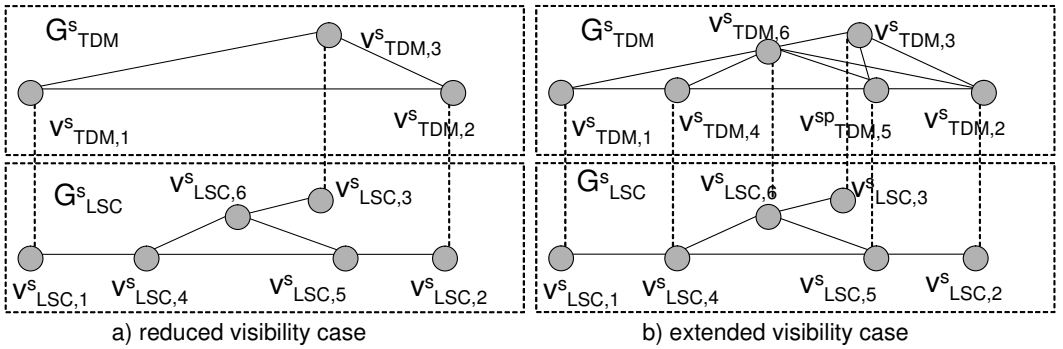

Fig. 5. Service Visibility Graphs (SVGs) (a) reduced visibility and (b) extended visibility. G^s_{LSC} is the same for both scenarios (full visibility).

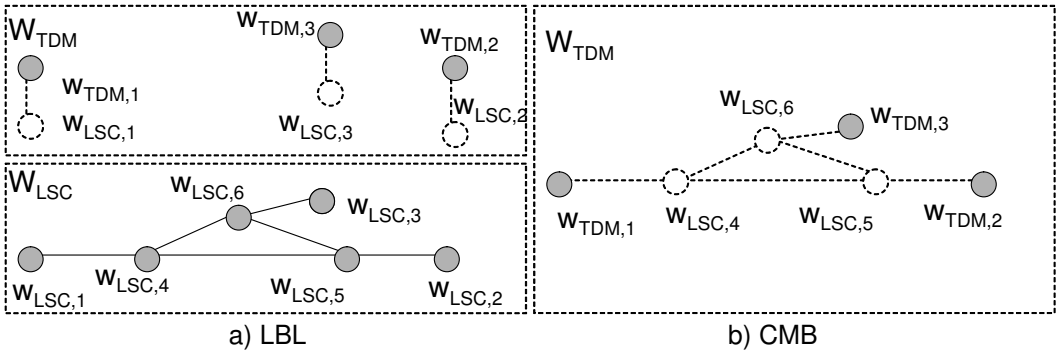

Fig. 6. Transformed graphs W_{TDM} and W_{LSC} with reduced visibility (a) $TransfLBL$ and (b) $TransfCMB$.

We illustrate now how the resource visibility concept can be used to facilitate sharing of a resource between two infrastructure services, e.g., S_1 and S_2 (Figure 7). Let us assume here that only the customer edge resources are TDM-switching capable. All provider switching resources, including the inter-layer resources are visible to both services S_1 and S_2. These services can use network links in the LSC layer (G^s_{LSC}) according to the visibility attribute assigned to each link. For example, a link between $v_{LSC,4}$ and $v_{LSC,5}$ is exclusively used by S_2. The edges in G^s_{TDM} denote the dynamic links. If in S_2 a dynamic link is created over P($e^s_{LSC,1,4}, e^s_{LSC,4,5}, e^s_{LSC,5,2}$), then its visibility will be $\{S_2\}$. If in S_2 a dynamic link is created over P($e^s_{LSC,1,4}, e^s_{LSC,4,6}, e^s_{LSC,6,5}, e^s_{LSC,5,2}$) its visibility will be $\{S_1, S_2\}$. This means that this dynamic link will be visible to both services and the information about a new FA will be exchanged between the control plane instances of the services. This scenario illustrate that resource visibility can be a powerful tool to dynamically virtualize the physical infrastructure and share the same resources by multiple infrastructure services.

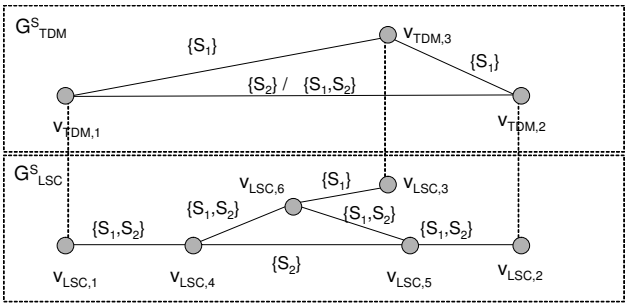

Fig. 7. Collaborative sharing or resource visibility information between two infrastructure services' control planes.

V. Performance Study and Numerical Results

In this section, we evaluate the performance of infrastructure service provision with GMPLS-TE using the concepts of resource visibility and GXP. The test networks and LSR architecture assumed in the analysis are depicted in Figure 8. All network nodes perform both wavelength switching (LSC) and TDM container switching and every LSR implements two switching layers, LSC and TDM. As shown in Figure 8, we assume that U_{TDM} is 4, i.e., one optical channel carries 4 STM-48 TDM containers (Figure 8d). The performance is evaluated within two main study cases: in the architecture with static domain interconnections (back to back, B-B) and in the architecture with GXP. The GXP topology assumes strictly non-blocking switching for any collocated domain or client. In all study cases, we use the resource visibility scenarios described earlier, and in particular the *reduced visibility* (RED) and *extended visibility*, (EXT). The link release strategies considered are *never release* (NR) and *release when idle* (REL) and the routing strategies used are *layer-by-layer* (LBL) and *combined* (CMB).

In all study cases, every traffic LSP request is TDM granular and on-demand; in the simulation, LSP inter-arrival times and duration are modeled according to negative exponential distribution. The requests are generally not symmetric, but each service member is equally probable to generate TDM-LSP requests to any other service member. All service members generate traffic with equal characteristics. The LSP requests

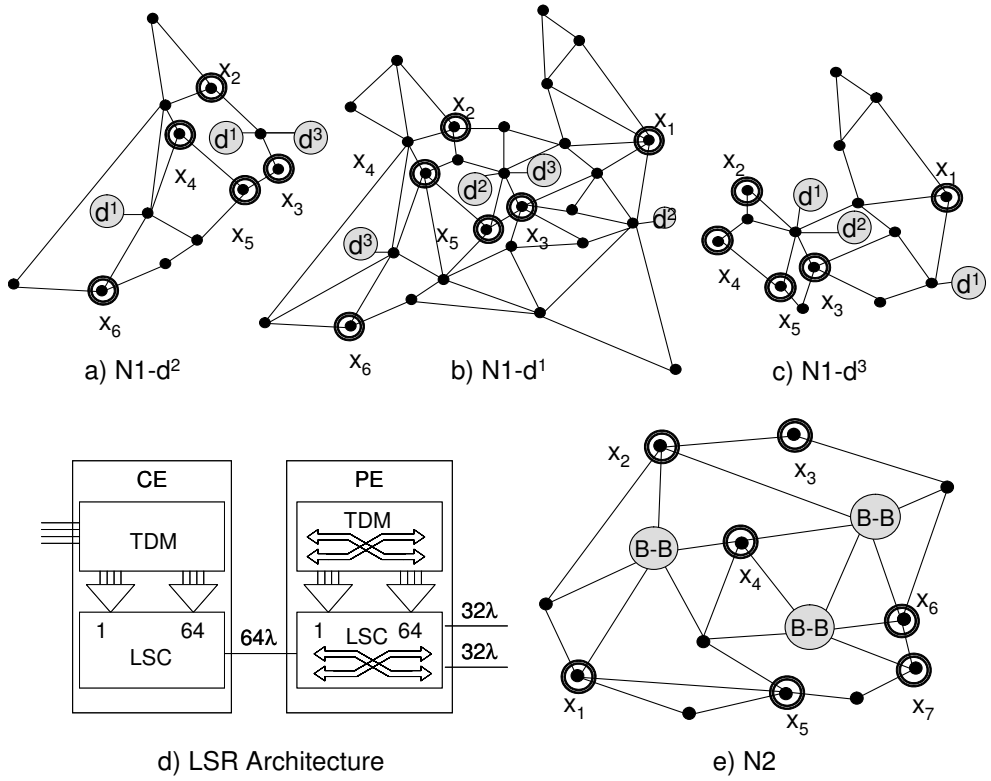

Fig. 8. Test networks and the LSR architecture used. Test network N1 includes three (partially overlapping) domains interconnected with static links at three points and dynamic links at six GRX. (a) Test network N1-domain d^2, GRX x_2, x_3, x_4, x_5, and x_6, and static interconnection links to domains d^1 at one node and to d^1 and d^3 at the other. (b) Test network N1, domain d_1, GRX x_1, x_2, x_3, x_4, x_5, and x_6, and static interconnections to d^2 and d^1 at three nodes. (c) Test network N1, domain d_3, GRX x_1, x_2, x_3, x_4, x_5, and static interconnection links to domains d^1 and d^2 (d) Two layer architecture of LSRs. (e) Test Network N2 with two overlapping domains, three static interconnections denoted by B-B or alternatively seven GRX instances denoted by x_1 to x_7.

that occur between domains (inter-domain) create only 10% of the total number of all requests generated. In the studies presented, the wavelength continuity constraint for LSC-LSP setup is not considered.

A. Resource Utilization and Impact of GXP

Figure 8(a,b,c) illustrates the first test network (N1) with 3 domains (d^1, d^2, d^3), 6 GXP nodes and 3 static links, based on the Pan-European network [28]. For this network, we study the efficiency of service traffic routing and TE resource management in terms of link utilization under a certain load condition. The range of blocking probability is kept low, i.e., below 5%. We assume here that all end-users belong to one single service and compare the distribution of the number of used LSC interfaces per link. The scenarios studied are defined as combinations of various visibility parameter (RED, EXT), TE release strategies (NR, REL), and routing approach (LBL, CMB). In the first study case we assume that the test topology deploys the static B-B links only and we study the efficiency of resource utilization measured by the link occupancy. This measure indirectly corresponds to the probability that a new connection request can be accommodated. In Figure 9, Y-axes show the percentage of all LSC resources with a link occupancy of 0%, $1-20\%$, $21-60\%$, $61-80\%$, $81-99\%$, and 100%. The link occupancy is measured in all domains. As shown in Figure 9, different strategies perform similarly relatively to each other within each domain. For example, scenario RED-LBL-NR shows a higher number of links with occupancy $> 60\%$ than RED-CMB-REL in all three domains. This is due to the fact that combined routing (CMB) unifies the resource information across all layers and can find a better path than LBL. It can be stated that CMB routing resembles a peer model between layers, since it constructs a connected multi-layer graph for routing. On the other hand, LBL routing resembles the overlay model since each layer makes an autonomous routing decision without considering resources in other layers. Therefore, LBL always shows a higher link occupancy than CMB. The capability to adapt to traffic changes, enabled through an efficient LSP release strategy (REL) brings considerable improvements in resource usage. Finally, applying an extending resource visibility (EXT) also brings improvements.

We now use the same test network to evaluate the performance impact of GXP. In this example, two different services are assumed, i.e., S_1 and S_2, within four test cases: M1, M2, M3 and M4. M1 and M2 are studied within the B-B topology while M3 and M4 assume the presence of GXPs. In the test cases M1 and M3, we assume that all five service members of service S_1 are internal to d^2. In absence of GXP, this service would be fully provisioned within d^2. With GXP, on the other hand, a domain internal service can be extended and provisioned over other domains when necessary. This is demonstrated in test cases M2 and M4, where S_1 extends over domains d^1, d^2 and d^3 and despite the fact that all service members are internal to one domain, the service is inter-domain. In all test cases, we also assume the second

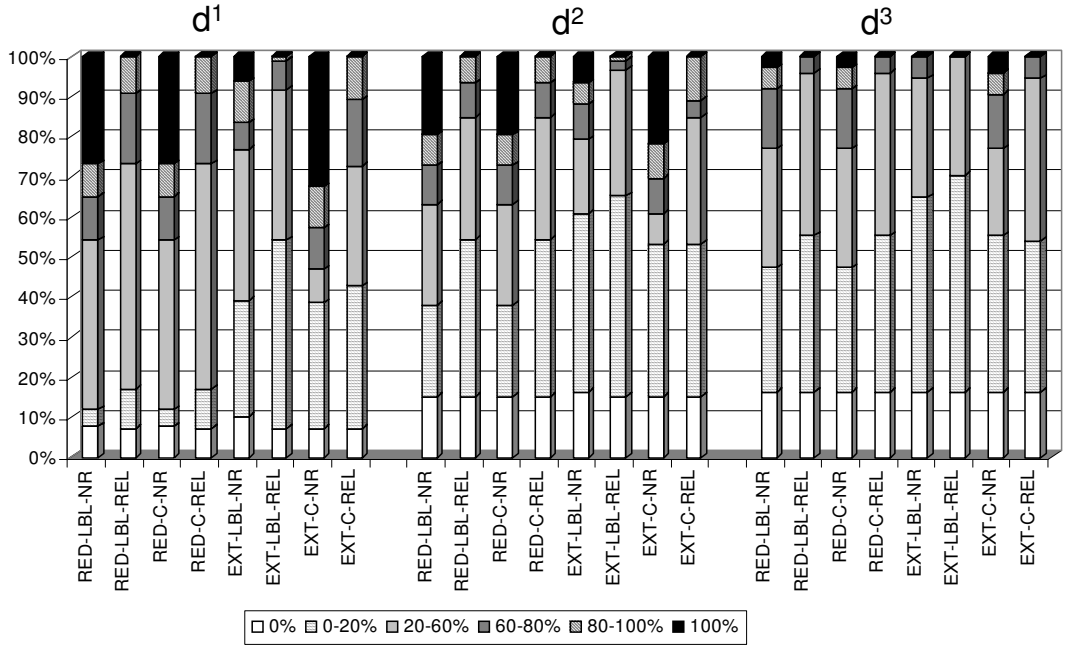

Fig. 9. Link Utilization. All links in N1 network, in domains d^1, d^2 and d^3 are shown sorted according to link occupancy.

service, S_2, provisioned to accommodate the so-called best-effort traffic with as many service members as there are nodes in all domains. The performance results for the four provision scenarios are presented in Table V-B, where we consider one specific best-effort traffic load (Erl=1) and three different service traffic loads ($L = 10Erl$, M=$20Erl$, H=$40Erl$), corresponding to network operation under low, medium and high service traffic load. From Table V-B, it can be seen that the service provision success significantly increases for all scenarios with extended visibility. The GXP-based services perform considerably better in general, especially for higher traffic loads. The S_1 success ratio (intra-domain service in d^2) is lower in the B-B topology due to the constraint that the service must be provisioned domain-internally. This is not the case in scenario M2 with GXP, where the performance significantly improves due to the inter-domain provider-provisioned service.

TABLE I
SUCCESS RATIO IN SERVICE TRAFFIC ROUTING. M1 AND M2 SCENARIOS USE N1 TOPOLOGY WITH STATIC B-B LINKS ONLY. M3 AND M4 SCENARIOS USE N1 TOPOLOGY WITH GXPS.

Scenario	Visibility	Link Release	Routing	Success Ratio					
				S_1			S_2		
				L	M	H	L	M	H
M1	RED	NR	LBL	99.7	86.7	48.0	100	99.5	77.0
M2	EXT	REL	CMB	99.9	94.2	62.0	100	99.7	83.8
M3	RED	NR	LBL	100	99.9	90.6	100	100	99.45
M4	EXT	REL	CMB	100	100	96.7	100	100	99.85

B. Multi-layer Resource Sharing

The concept of visibility introduced in this paper enables the provider to selectively offer resources to services. In the case of reduced visibility, each service is offered only wavelength pipes, of which the utilization can only be controlled service internally. Extended visibility enables an adaptation to the requirements of the traffic, by allowing service members to connect with each other at any granularity. In this case, since multiple services use and control parts of the same multi-layer resource within the network, the management of every shared resource becomes a challenge. In addition, the network provider driven TE becomes more complex, since the TE decisions as of which resources are to set up and release can not be tailored to one service, i.e., they impact multiple services. Nevertheless, if the visibility information is shared between the provider and service control planes, the TE decisions can be taken collaboratively, and services can tailor their routing schemes to the requirements of other services. Such a model can also be developed with economic incentives in mind.

We next illustrate the results related to such a collaborative resource sharing. The test network is shown in Figure 8e, and comprises of two provider domains with 15 LSRs per domain and 7 GXP instances, placed according to the GXP placement heuristics presented in [13]. Each intra-domain and inter-domain link contains 32 and 124 wavelengths, respectively. We define service S_1 so that the service members (total 7) are all within the domain d^1, and also collocated with GXPs. We also consider the best-effort traffic randomly generated with a low load, and denote this traffic as service S_2. The other assumptions are the same as in the previous test cases. The routing method applied here is CMB. Figure 10 shows the LSP blocking probability of both services in B-B architecture, for reduced and extended visibility case, and for different link release strategies. Contrary to the previous results, the best performance in this scenario is achieved where services establish direct wavelength pipes (RED) and release them adaptively to the traffic. In other words, a wavelength pipe is released as soon as it is not utilized anymore (REL). We analyzed these and other scenarios and found that the reason why EXT scenarios do not always perform better than

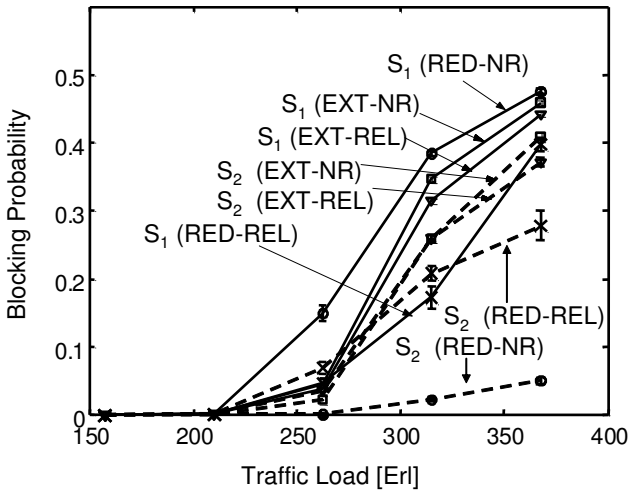

Fig. 10. Blocking Probability for different traffic loads. S_1 is a domain-internal service, and S_2 is a random, best-effort service.

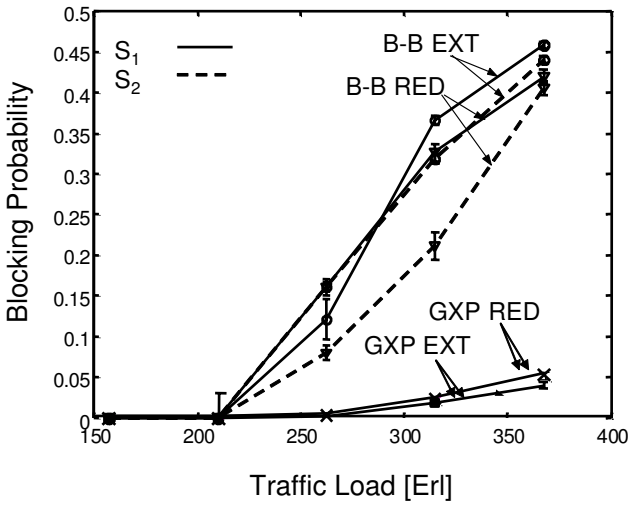

Fig. 11. Blocking probability in two services scenario with GRX and B-B architecture.

RED is in the fact that with EXT there may be too much of uncontrolled (greedy) resource sharing. In our particular case, the random best effort traffic and the service traffic did not benefit from sharing resources without coordination. The collaborative and coordinated sharing of information based on visibility can alleviate the problem. In this case, a service can change visibility of a specific dynamic link based on some knowledge of the topology it is trying to build.

In the last example, we consider two services: S_1, defined as in the previous example, and S_2, an inter-domain service with three service members connected to d^1 and four to d^2. Based on the results shown in Figure 10, we now compare the performance of S_1 and S_2 with B-B and GXP architectures. Figure 11 confirms that the GXP architecture brings significant improvements. With GXP, service can use any available domain. For higher loads, EXT seems to be less efficient as compared to RED, particularly without GXP. In other words, when the network is operating with higher service loads, the exclusive usage of resources is more efficient. A number of additional case studies not shown here confirm that although multi-layer routing and resource allocation may seem to be a better strategy, it may require a service-aware resource allocation with the dedicated resources to deliver a guaranteed service performance.

VI. Summary and Conclusions

In this paper, we analyzed the performance of infrastructure service provisioning with two novel GMPLS TE concepts: resource visibility and inter-domain exchange. Resource visibility is a new network control plane concept, which defines the usage polices for transmission, multiplexing, and switching in multiple GMPLS layers. The inter-domain exchange, referred to as GMPLS Exchange Point (GXP), is the physical layer equivalent of the Internet Exchange Point (IXP). We modeled the dynamic provisioning of infrastructure services using graph theory and deployed GMPLS TE with resource visibility and GXP to optimize the routing and resource yields.

The results show that the majority of performance improvements can be obtained with a controlled usage of multi-layer resource visibility and with a more flexible interconnection architecture between domains. We found that significant efficiency can be achieved in resource utilization (up to 30%) in cases where extended visibility allows for a more adaptive resource allocation, tailored to traffic requests. However, we also found that extended visibility of resources deserves a careful consideration, and coordination of service-specific TE actions. Without coordination, it may happen that a greedy TE with extended visibility results in poorer performance with respect to a controlled TE, i.e., with reduced visibility. Specifically, we found that extended visibility brings benefits to both customers and providers only when LSPs are setup as a result of collaborative traffic engineering policies, which means sharing of information relative to the resource capabilities and utilization. When a collaborative TE is not possible or not deployed, the single layer TE can outperform the multi-layer TE.

A number of remaining related issues require further investigation. For example, how will TE strategies with resource visibility scale with multiplicity of GMPLS layers, network protocols and services? Will the performance improvement achieved through GXP-like interfaces justify their introduction between domains? How can GXP impact trusted visibility mediation and resource trading within the physical layer? How can resource providers become interested in exploiting the opportunities of opening up their infrastructures, and in profiting from economic benefits of collaborative resource allocation? Although the answers to these questions may depend on specific future network technologies and business models, the GMPLS TE strategies introduced in this paper are the first steps towards an efficient and carrier-neutral infrastructure provision in network operation practices.

Acknowledgment

The authors would like to thank anonymous reviewers for their insightful comments.

REFERENCES

[1] Grid Networks: Enabling Grids with Advanced Communication Technology F. Travostino (Editor), J. Mambretti (Co-Editor), G. Karmous-Edwards (Co-Editor) Wiley Publishers September, 2006

[2] CA*net User Controlled Lightpah, UCLP v2, http://grid2.canarie.ca/wiki/index.php/UCLPv2

[3] NSF Initative GENI, http://www.nsf.gov/cise/geni/

[4] T. Takeda, et. al.: "Layer 1 Virtual Private Networks: Service Concepts, Architecture Requirements, and Related Advances", IEEE Commun. Mag., Mar 2004.

[5] I-NEC: "Declaration of Open Networks," www.i-nec.com, Nov. 2006.

[6] C. Metz: "Interconnecting ISP networks" IEEE Internet Comput., Volume: 5 Issue: 2, March-April 2001 Page(s): 74 -801

[7] H. Ould-Brahim, et. al.: "GVPN: Generalized Provider-provisioned Port-based VPNs using BGP and GMPLS, draft-ouldbrahim-ppvpn-gvpn-bgpgmpls-01.txt, Dec. 2002.

[8] H. Ould-Brahim et al.: "Service Requirements for Optical VPNs", draft-ouldbrahim-ppvpn-ovpn-requirements-01.txt, Jul 2003.

[9] D. Awduche, J. Agogbua, J. McManus: "An Approach to Optimal Peering Between Autonomous Systems in the Internet," Proc. of International Conference on Computer Communications and Networks, 1998.

[10] I. Nakagawa, et al "A design of a next generation IX using MPLS technology", Applications and the Internet, 2002. (SAINT 2002). Page(s): 238 -245.

[11] Blyth, K.J.; Cook, A.R.J.: "Designing a GPRS roaming exchange service," Conference on 3G Mobile Communication Technologies, 2001, pages 201 -205.

[12] B. St. Arnaud, et. al.: "BGP Optical Switches and Lightpath Route Arbiter" Optical Networks Magazine Mar/Apr 2001.

[13] S. Tomic, A. Jukan: "MPFI: The Multi-provider Network Federation Interface for Interconnected Optical Networks", Proc. Globecom 2002, Nov. 2002, Taipei, Taiwan.

[14] F. Dijkstra, C. de Laat: "Optical Exchanges", Proc. GRIDNETS 2004, Oct 2004.

[15] D. Awduche, et al.,: "Requirements for Traffic Engineering Over MPLS", RFC2702, Sep 1999.

[16] R. Sabella, et. al.: "A Multilayer Solution for Path Provisioning in New-Generation Optical/MPLS Networks", J. Lightwave Technol., May 2003, pp. 1141-1155.

[17] H. Zhu, H. Zang, K. Zhu and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," IEEE/ACM Trans. Networking (TON), vol. 11, no. 2, pp. 285-299, April 2003.

[18] A. Gencata, B. Mukherjee: "Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic", IEEE/ACM Trans. Networking, vol. 11, no. 2, April 2003.

[19] S. Tomic, A. Jukan:" Policy-based Lightpath Provisioning over Federated WDM Network Domains", Proc.OFC 2002.

[20] Y. Zhu, W. Alanqar, A. Jukan, M. Ammar: "End-to-End Service Provisioning in Multi-granularity Multi-domain Optical Networks", Proc. ICC 2004, Jun 2004, pp. 1579-1583.

[21] V. Prevelakis, A. Jukan: "How to buy a network: trading of resources within the physical layer", IEEE Commun. Mag., December 2006, to appear.

[22] G. Karmous-Edwards: "Global E-Science Collaboration", IEEE Comput. Sci. Eng. Mag., March/April 2005.

[23] K. Kompella, et. al.: "Multi-area MPLS Traffic Engineering", work in progress draft-kompella-mpls-multiarea-te-05.txt, Oct 2003.

[24] K. Kompella, Y. Rekhter: "LSP Hierarchy with Generalized MPLS TE", RFC 4206, draft-ietf-mpls-lsp-hierarchy-09.txt, Oct 2005.

[25] A. Farrel, J.-P. Vasseur, J. Ash, "A Path Computation Element (PCE)-Based Architecture", IETF RFC 4655, Informational,August 2006.

[26] K. Kompella, et. al.: "OSPF Extensions in Support of Generalized MPLS", RFC 4203, draft-ietf-ccamp-ospf-gmpls-extensions-13.txt, exp. Oct 2005.

[27] E. Mannie, et al.: "GMPLS Architecture", RFC 3945, draft-ietf-ccamp-gmpls-architecture-04 work in progress, Oct 2004.

[28] S. Tomic, A. Jukan: "Dynamic Provisioning over Multi-provider Interconnected GMPLS-enabled Networks", Proc. ONDM 2003, Feb. 2003, pp. 365-383.

Slobodanka Tomic graduated from the Faculty of Electrical Engineering in Belgrade, Yugoslavia. In November 2000 she joined the Institute of Broadband Communications at Vienna University of Technology in Austria as a researcher working towards a Ph.D. Prior to that she was working in the industry where she engaged in several research and development projects focusing on network control and management for ATM, SDH and WDM networks. In particular, within the project ACTS AC231 Management of Optical Networks (MOON), Ms. Tomic was working on architectural and network management concept for European wavelength-routed network field trial. Since September 2006, Ms. Tomic is with The Telecommunications Research Center Vienna (ftw.), Vienna, Austria. Her research interests include protocols and architecture for new generation ubiquitous networks and services.

Admela Jukan received the B.Sc. from the University of Zagreb, Croatia, M.Sc. degree in Information Technologies and Computer Science from the Polytechnic of Milan, Italy, and the Ph.D. degree (cum laude) in Electrical and Computer Engineering from the Vienna University of Technology (TU Wien), Austria. She is currently Professor at Institute Nationale della Recherche Scientifique of University of Quebec and also Visiting Professor at the University of Illinois at Urbana Champaign (UIUC). Prior to joining UIUC, she served as Program Director in Computer and Networks System Research at the National Science Foundation (NSF) in Arlington, VA, and she was Research Assistant Professor at the Georgia Institute of Technology in Atlanta, GA. While at NSF, she was responsible for funding and coordinating US-wide university research and education activities in the area of network technologies and systems. In 1999 and 2000, she was Visiting Scientist at Bell Labs, NJ. Dr. Jukan is the author of numerous papers in the field of networking, and she has authored one and edited several books. She is a recipient of numerous research awards in Europe and in the US. Dr. Jukan has chaired and co-chaired several international conferences, including ONDM, ICC and GLOBECOM.