A Service Oriented Framework for Efficient Resource Orchestration in Future Optical Networks

Chinwe E. Abosi, Reza Nejabati, Dimitra Simeonidou
School of Computer Science and Electronic Engineering
University of Essex, Colchester, UK, CO4 3SQ
Email: {ceabos, rnejab, dsimeo}@essex.ac.uk

Abstract—**The future Internet evolution is driven by applications that require simultaneous real-time access to multiple heterogeneous IT (e.g. computing and storage) resources interconnected by high-speed optical networks. Service oriented architectural frameworks have been considered to improve service delivery in this environment by organising distributed heterogeneous resources owned by multiple providers.**

In this paper, we discuss a service oriented architectural framework tailored to the needs of future Internet applications. The framework introduces the capability to orchestrate the provisioning of distributed heterogeneous resources owned by multiple providers. Central to the framework is a novel resource orchestration model. This model focuses on the optimized selection of heterogeneous IT and network resources owned by different Infrastructure Providers (InPs). The proposed model aims to achieve a global optimum from both the end-users' and InPs' point of view across different administrative domains. An integer linear program (ILP) is formulated to obtain optimal results that maximizes the number of accepted requests while minimising resource usage. It is compared against a co-scheduling ILP model, whose objective is to maximise the number of accepted requests only. We then propose a heuristic solution for scalability. Its performance is compared against (i) our proposed ILP (ii) a co-scheduling heuristic that aim to maximise the number of accepted requests only and (iii) an algorithm that does not take into account cross-domain optimisations. Finally, two online algorithms for operational scenarios is proposed which outperforms three benchmark algorithms.

I. Introduction

Profound changes in emerging application models and the convergence of network and IT (e.g., storage, computational, and content) resources have fuelled the need to enhance the current Internet architecture and its operation [1]–[4]. These changes include the introduction of high-end IT-based applications that require computational, storage and/or visualisation equipment across the network [5]–[9]. In addition, the current development and technical enhancement of photonic network technologies [1], [10]–[12], dynamic control planes [13]–[16], distributed multi core processing [17]–[20] and distributed data repositories [21]–[26] are contributing to the profound transformation of users capabilities.

As a result of these advances in technology and the increase in user capabilities and expectations, more demanding IT-based applications such as cloud computing, interactive 3D games, networked media and high definition digital cinema are emerging. These applications require high bandwidth to transfer large amounts of data, Quality of Service (QoS) guarantees as well as IT services. Thus, the future network can be said to be characterised by the global delivery of high-performance IT services over a high-capacity network that is able to provide high levels of QoS. The bandwidth and QoS requirements can be achieved by establishing dedicated wavelength paths between end sites, called lightpaths over a wide-area fibre optic Wavelength Division Multiplexing (WDM) network [27]. The WDM network has numerous favourable characteristics such as its global reach, high speed, and huge bandwidth capacity, making it a natural choice to provide the transport platform needed to support these emerging high-end Internet applications [27], [28].

Today's Infrastructure Providers (InP) are now facing an increasing need to provide users with dynamic high-capacity and high-performance optical network connectivity services tightly bundled with IT processing and storage services at IT resource sites. To achieve this, InPs need to incorporate external partners or administrative domains, called Third Party InPs, to enhance their service offerings to their users [2], [29]–[31]. That is, providers that offer network services will need to collaborate with providers that offer IT services such as computational and/or storage services.

This is almost impossible without a well-developed model that can manage and benefit from the integration of network and IT technologies as well as disparate, heterogeneous InPs, and users. However, the heterogeneous nature of the underlying resource infrastructure (i.e. computational vs. storage vs. network bandwidth) plus the competitive relationship that may exist between InPs can make providing the type of end-to-end services needed by these emerging applications a daunting task.

To realize this kind of optically networked-IT infrastructure environment, and achieve the level of collaboration required, a new architectural framework is required within the current Internet architecture. The proposed framework will need to support the orchestrated use of optical network and IT resources. It will include mechanisms to coordinate Network InPs (NInP), Storage InPs (SInP), Compute InPs (CInP) and Content InPs (COInP) to maximise the utilisation of InP resources and minimise the probability of blocking user's requests. In addition, this framework must seamlessly integrate optical network technologies and IT resources, and provide a

Manuscript received August 14, 2011; revised October 12, 2011; accepted November 8, 2011.

© 2011 ACADEMY PUBLISHER
doi:10.4304/jcm.6.9.711-723

mechanism for holistic and efficient service delivery.

The above factors are the motivation behind a service oriented architectural framework suitable for the future Internet. The Organization for the Advancement of Structured Information Standards (OASIS) [32], define a service oriented architecture as a *"paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains ... to produce desired effects consistent with measurable preconditions and expectations."* Thus, the aim of the framework is to match the needs of users with heterogeneous resources across multiple administrative domains that address their needs. The proposed service oriented architectural framework allows the coordinated consideration of network and IT resources (e.g., storage, computational, and content) to provide the networked-IT services required by emerging applications and their users.

In the following sections, we introduce the proposed service oriented architectural framework. We then focus on the resource orchestration algorithms which are implemented to determine an optimised mapping of user's request to InP resource capability. Resource orchestration algorithms were first introduced by the authors in [33]. The aim of these resource orchestration algorithms are to select appropriate resources from multiple InPs to improve service provisioning in two ways - (i) maximising accepted requests while (ii) minimising resource usage in terms of number of hops. The algorithms ensure that when the task scheduling and routing and wavelength assignment is implemented on the selected resources by the individual underlying infrastructure's control and management systems, a global optimum across all the domains can be achieved. In [33], the authors investigated this concept under offline scenarios.

This paper extends the work in [33] by positioning resource orchestration within the framework of the service oriented architecture as defined in [32], [34]. Furthermore, the online scenario is also considered, in addition to the offline scenario presented in [33], to investigate the performance of the resource orchestration in operational, on demand resource provisioning situations. Thus, we formulate and present both offline and online resource orchestration algorithms. The resource orchestration problem is first formulated as an Integer Linear Programming (ILP) model. We compare the proposed ILP with an ILP formulation that aims to maximise the number of accepted requests. A near-optimal heuristic approximation algorithm is also presented for scalability and compared to the ILP formulation over a small 5-node network. The heuristic is then studied over the larger European Optical Network (EON) and the NSF network and compared against an algorithm that attempts to sequentially locate and select resources from individual InP domains. Finally, to deal with on-demand operational scenarios, several on-line orchestration algorithms are proposed and evaluated.

II. Architectural Framework Description

In this section, we introduce the Service Oriented Architecture for the Future Internet (SOAFI) framework capable of organising resources from different administrative domains to provide enriched future Internet services [34].

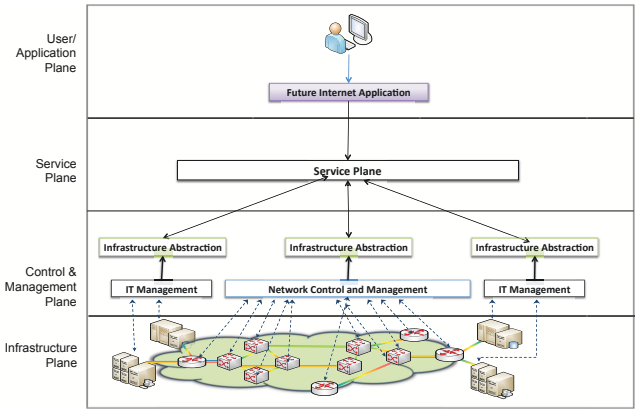


Fig. 1. Architectural Overview

Fig. 1 presents the key features of SOAFI. The SOAFI framework is designed to support the integration of multiple InPs required to provide better service delivery performance in terms of reduced blocking and improved resource usage. It includes: (i) *abstraction algorithms* that allows InPs to summarise and hide details of their infrastructures. It encourages InPs to share their resource information, which is required for the orchestration of services. Several abstraction algorithms suitable for this entity were proposed and evaluated in [34], [35]; (ii) a *semantic description framework* to ensure that there is uniformity in the way resources and requests are described to facilitate automatic, autonomic matching of needs and capabilities [36]; (iii) a *marketplace* which acts as a global database to store the resource information from multiple InPs and (iv) *orchestration algorithms* that use the global information in the marketplace to optimally select and match the best InPs and their respective resources to compose the required services needed to satisfy each users need. Moreover, the proposed framework has been designed such that it exists in a separate layer to decouple the matching of needs and capabilities from the advancement of the application and infrastructure layers. This allows for each of these layers to evolve independently allowing for seamless deployment of future optical Internet services, while supporting existing services.

These characteristics of the SOAFI framework are encompassed within two key entities namely: (i) infrastructure abstraction entity, and (ii) the Service Plane (SP) as shown in Fig. 1. The SP encompasses the semantic description framework, the marketplace and the orchestration engine. The abstracted resource information is stored in the marketplace entity as shown in Fig. 2. The orchestration algorithms are then used to coordinate the interaction between network and IT resources and the InPs that provide, control and manage these resources in order to compose emerging networked-IT services. Thus, the SP aims to seamlessly coordinate the interaction-between and access-to both network and IT resources in order to facilitate a better and more efficient service

© 2011 ACADEMY PUBLISHER

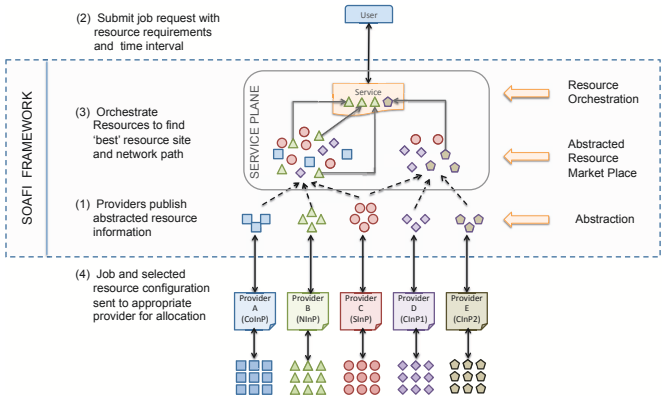


Fig. 2. Procedure of Dispatching a User's Request in the SOAFI Framework

provisioning globally and within the individual domains of each InP. The SP (i) acts as a "matchmaker" between users and the multiple InPs that may have the resources they require and (ii) creates a collaborative environment that facilitates the interaction between both network and IT resources and the InPs that provide, control and manage these resources.

Resource orchestration refers to the locating, coordinating and selecting of resources from various InPs to fulfil a users requirement. It aims to select the 'best' or most optimal IT resource site and network path to this selected IT resource site to satisfy a job request. The selected resource information is sent to the control and management entities of the respective InPs (the local resource management systems (LRMS) of IT resources and Network Resource Provisioning System (NRPS) of the optical network) through a standard interface. InPs then use this information to perform low-level task scheduling and lightpath and wavelength assignment accordingly.

The workflow and interactions within the SOAFI framework when dispatching a users request is illustrated in Figure 2. The InPs first publish abstracted representation of their IT and optical network resource information to the SP. A user submits a request to the system via a user's portal. The resource orchestration engine in the SP receives the user's request and accesses the marketplace to locate, and select appropriate resources from amongst the different InPs. The result is passed back to the relevant InP to provision and execute the job.

III. Related Works

The closest research topic to resource orchestration as implemented within the SOAFI framework is the joint or co-scheduling of resources. This involves the scheduling of network and IT resources simultaneously to support distributed applications. It normally involves a broker, which exists within either the middleware, management or control planes, tasked with joint scheduling of resources. The broker also implements low level task scheduling and routing and wavelength assignment [37], [38] to select, schedule and allocate required resources.

Meta-Scheduling Systems (MSS) that implement joint scheduling of resources, such as the Data Intensive And Network Aware (DIANA) environment introduced in [39] and the Joint Resource Scheduling System (JRSS) introduced in [40], are closely related to the proposed SP resource orchestration. Like the SP, the authors in [40] present a joint meta-scheduling system that co-allocates network and IT resources in a separate, yet integrated entity. These meta-schedulers use co-scheduling algorithms [41]–[44] to jointly select network and IT resources. [41] aims to maximise the number of accepted request while minimising an arbitrary cost based on the requested capacity to execute the request. [42] considers the maximisation of the economic benefit to InPs by introducing an economic cost for using a resource and a budget a client is prepared to pay for accessing the resource. [43] presents two separate objective functions to maximise the number of accepted request and minimise the number of wavelengths used. The authors in [44] propose a joint scheduling model which first sorts the jobs to be scheduled into a list according to some defined priority of each job and then schedules each job to a specific resource.

Like MSSs, the SP facilitates the requesting of resources from more than one InP, and schedules the required resources across these InPs. Unlike the DIANA and JRSS MSS proposals, which implements resource management, low-level task assignment and application execution; resource orchestration performs a high-level co-scheduling of resources using abstracted resource information. A site and path is selected by the resource orchestration algorithm, while the actual allocation of the job on individual processors, diskspace locations or wavelengths is left to be implemented by the LRMS and NRPS of the underlying infrastructure.

IV. Resource Orchestration

The SP presents a platform through which resources from different providers can be coordinated, orchestrated and selected in a single-step process. The orchestration process performs a high level joint scheduling of resources. Each request may require a number of reservations. The orchestration algorithms need to: (i) select an appropriate provider(s) to provision the requested service (resources). In some cases, there is more than one provider with sufficient resources to satisfy the request. This provider is either one of a Storage InP, Computational InP, Content InP, Network InP or a combination thereof, depending on the user's requirements. (ii) select a possible path on which to provision the request if a Network InP was selected in Item (i) above. The selected path is communicated to the Network InP as a suggested path. The Network InP may use another path in provisioning the request if the selected path on the physical infrastructure does not have sufficient capacity to satisfy the request. We recall that the decision as to which wavelength to use and which lightpath to groom the request onto is performed by the control plane employed by individual Network InPs. In the same way, the IT management utilised by the Computational InP and Storage InP is responsible for the low level node and task scheduling at each IT resource site.

© 2011 ACADEMY PUBLISHER

A. System Model

This section presents a model for the infrastructure. It includes: (i) IT resource nodes, $\mathcal{R}$; and (ii) users that generate job requests, interconnected to (iii) an optical mesh network. IT resource sites offer either data storage resources, $S \subseteq \mathcal{R}$, computational resources, $C \subseteq \mathcal{R}$, or video servers, $\Phi \subseteq \mathcal{R}$, containing files that need to be streamed to the requester. Each compute site, $c \in C$ is described by its residual processing capacity, P_c, in MIPS. Each storage sites, $s \in S$, has a finite capacity represented by D_s. Each video server, $\phi \in \Phi$, can serve a maximum number of concurrent sessions. IT resource sites, $\mathcal{R}$, and users are connected to the network through opaque OXCs with optical bypass. The network is an optical circuit switched network consisting of N nodes, $\mathcal{V}$, connected by optical fibre links, $\mathcal{E}$. Each link, $e_{(u,v)}$ has W wavelengths, each with a finite transmission capacity $B^w_{(u,v)}$. We assume that each OXC is equipped with unlimited wavelength conversion and traffic grooming capabilities. In this way, a lightpath can be established between any source-destination pair as long as there is enough bandwidth resources on the links in the path. Video servers are assumed to have the requested files, allow unlimited concurrent sessions, and satisfy all media-related constraints, hence the only constraints we consider for these requests is link bandwidth. To simplify the problem, we assume that the links connecting IT resources and users to the network have infinite bandwidth. Hence we can neglect this connection in our model.

There are two types of job requests, $j \in J$. Type A requests are either computational, $J^{A_{comp}}$ or storage $J^{A_{stor}}$ requests. They are associated with a 5-uple $(src_j, b_j, r_j, t_j^{start}, t_j^{end})$ where $src_j \in V$ is the source; b_j is the required bandwidth. For computational requests, data is sent on output links (ε^{out}) to a computation site, processed at the site and the resulting data is returned on input links (ε^{in}) to the requester. For storage requests, data is sent on output links (ε^{out}) to a storage site. r_j is the requested IT resource capacity equal to p_j if the requested resource type is computational and d_j for storage requests. $[t_j^{start}, t_j^{end}]$ is the time interval during which the job is active. Type B requests are video requests, represented by a 5-uple $(src_j, \phi_j, b_j, t_j^{start}, t_j^{end})$, where $src_j \in V$ is the requesting node; $\phi_j \in \mathcal{R}$ is the id of the streaming server node. For this request type, bandwidth is scheduled on input links (ε^{in}) to mimic the streaming of files from the server to the requester. b_j and $[t_j^{start}, t_j^{end}]$ are as previously defined.

B. Scenarios

To evaluate our resource orchestration algorithms, we consider two scenarios:

1) In the first scenario, the set of job requests are known in advance. Thus the objectives can be presented and formalised using ILP models. However, ILPs have exponentially increasing run-times as the network and job matrix size increases. For this reason, several heuristic approximation algorithms are proposed and implemented to solve the orchestration problem in more scalable environments. Because of the nature of heuristic approximation algorithms which require that the set of job requests are known in advance, near optimal decisions can be made as to which request to accept/reject to optimise resource usage as well as job acceptance.
2) The second scenario presents a typical on-demand operational scenario. In this scenario, the characteristics of jobs are only known at the time that they require resources. To elaborate: at a given time, t, a job, j_{t+x} which is to be scheduled at time $t + x$, $(x > 0)$ is not known. This is regardless of if the job, j_t which is to be scheduled at time t and the job, j_{t+x} are time-overlapped (i.e. at some time, the jobs will need to run concurrently). For this type of jobs, on-line orchestration algorithms are required.

C. ILP Model Formulation

In this section, we introduce an Integer Linear Programming, ILP, formulation for the resource orchestration problem and an approximation heuristic for scalable evaluation.

To formalise the ILP, we introduce the following notations and binary variables:

- θ_j is a binary variable, $\theta_j \in [0, 1]$, where $\theta_j = 1$ if job j is accepted. $\theta_j = 0$ otherwise.
- $\varepsilon^{in/out}_{j,(u,v),t}$ is a binary variable, $\varepsilon^{in/out}_{j,(u,v),t} \in [0, 1]$, where $\varepsilon^{in/out}_{j,(u,v),t} = 1$ if job j uses link (u, v) for input/output data at time instance, $\tau_t \in [t_{start}, t_{end}]$. $\varepsilon^{in/out}_{j,(u,v),t} = 0$ otherwise.
- $\gamma_{j,c,t}$ is a binary variable, $\gamma_{j,c,t} \in [0, 1]$. $\gamma_{j,c,t} = 1$ if job j uses a compute node at an IT resource site c for processing data at time instance, $\tau_t \in [t_{start}, t_{end}]$. $\gamma_{j,c,t} = 0$ otherwise.
- $\delta_{j,s,t}$ is a binary variable, $\delta_{j,s,t} \in [0, 1]$. $\delta_{j,s,t} = 1$ if job j uses a storage node at an IT resource site s for storing data at time instance, $\tau_t \in [t_{start}, t_{end}]$. $\delta_{j,s,t} = 0$ otherwise.

Our objective, given in Equation 1, is to ensure that the underlying network resources are used efficiently in satisfying job requests. We consider only network resources in this objective function. This is because a single job request can use more than one network link to satisfy its requirement. In this case, the requested bandwidth is replicated on each of the links. However, each job request can be satisfied on only a single IT resource or none at all. Thus our system model does not allow for the possibility of resource wastage on IT resources which can occur on network resources.

$$RU = \min \sum_{e_{(u,v)} \in \mathcal{E}, j \in J} \varepsilon_{j,(u,v)} \tag{1}$$

The set of constraints is the following: The first constraints, Equations (2) and (3), ensure that the maximum number of requests can be accepted.

$$\sum_{j \in J} \theta_j \geq ACC \tag{2}$$

© 2011 ACADEMY PUBLISHER

where ACC represents the maximum number of requests that can be accepted. If there is sufficient capacity to accept all the job requests, Equation (2) transforms to:

$$\sum_{j \in J} \theta_j \geq |J| \tag{3}$$

$$\sum_{j} p_j \cdot \gamma_{j,c,t} \leq P_c, \quad \forall c \in C \; \forall t \in T \tag{4}$$

$$\sum_{j} d_j \cdot \delta_{j,s,t} \leq D_s, \quad \forall s \in S \; \forall t \in T \tag{5}$$

$$\sum_{j} b_j \cdot \varepsilon^{in}_{j,(u,v),t} \leq B^w_{(u,v)} \cdot W, \quad \forall e_{(u,v)} \in E \; \forall t \in T \tag{6}$$

$$\sum_{j} b_j \cdot \varepsilon^{out}_{j,(u,v),t} \leq B^w_{(u,v)} \cdot W, \quad \forall e_{(u,v)} \in E \; \forall t \in T \tag{7}$$

Equations (4) - (7) are capacity constraints. They state that any job request that is to be satisfied at processing site, c, storage site, s, and/or link, $e_{(u,v)}$ at time t must not exceed the maximum capacities available for each type of resource.

$$\sum_{c} \gamma_{j,c} = \theta_j, \quad \forall j \in J^A_{comp} \tag{8}$$

$$\sum_{s} \delta_{j,s} = \theta_j, \quad \forall j \in J^A_{stor} \tag{9}$$

$$\varepsilon^{in}_{j,(u,v)} \leq \theta_j, \quad \forall j \in J \; \forall e_{(u,v)} \in E \tag{10}$$

$$\varepsilon^{out}_{j,(u,v)} \leq \theta_j, \quad \forall j \in J \; \forall e_{(u,v)} \in E \tag{11}$$

Equations (8) - (11) are the resource allocation constraints. (8) and (9) force each storage or computational job request to use only one storage site or one computational site respectively based on its requirement. Bandwidth allocation is forced by (10) and (11) to use as many links as needed for each accepted job request.

$$\forall j \in J \; \forall u \in V$$

$$\sum_{\substack{v \in V \\ e_{(u,v)} \in E}} \varepsilon^{in}_{j,(u,v)} - \sum_{\substack{v \in V \\ e_{(v,u)} \in E}} \varepsilon^{in}_{j,(v,u)} = \begin{cases} +\theta_j & \text{if } u \mapsto \Phi \\ +\gamma_{j,u} & \text{if } u \mapsto C \\ -\theta_j & \text{if } u \mapsto C, \Phi \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\forall j \in J \; \forall u \in V$$

$$\sum_{\substack{v \in V \\ e_{(u,v)} \in E}} \varepsilon^{out}_{j,(u,v)} - \sum_{\substack{v \in V \\ e_{(v,u)} \in E}} \varepsilon^{out}_{j,(v,u)} = \begin{cases} +\theta_j & \text{if } u \mapsto C, S \\ -\gamma_{j,u} & \text{if } u \mapsto C \\ -\delta_{j,u} & \text{if } u \mapsto S \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The flow conservation for the network resources are forced by Equations (12) and (13). The equations ensure that the correct direction of flow for each job request is as detailed in Section IV-A. Furthermore, the equations also ensure that if a job, j is accepted, then all required resource types for job j are allocated or none of the resource type is allocated. The right side of both equations ensure that the correct type of resource is used to satisfy a requirement. The notations, $u \mapsto C, u \mapsto S, u \mapsto \Phi$, associate compute and storage resources as well as servers with the edge nodes of the network.

$$\varepsilon^{in}_{j,(u,v)} + \varepsilon^{in}_{j,(v,u)} \leq 1 \qquad \forall j \in J \; \forall e_{(u,v)} \in \mathcal{E} \tag{14}$$

$$\varepsilon^{out}_{j,(u,v)} + \varepsilon^{out}_{j,(v,u)} \leq 1 \qquad \forall j \in J \; \forall e_{(u,v)} \in \mathcal{E} \tag{15}$$

Equations (14) and (15) avoid loop formation by ensuring that a link, $e_{(u,v)}$ is not used in both directions to satisfy a job request.

D. Offline Orchestration Heuristics

We introduce a Modified Simulated Annealing (MSA) heuristic for the orchestration of resources since ILPs cannot solve large complex problems in realistic times. Our objective is to minimise the number of hops used in accepting the maximum number of job requests. The MSA algorithm is based on the Simulated Annealing (SA) algorithm which is a generic probabilistic meta-algorithm used to solve global optimization problems [45]. The pseudo-code of the MSA

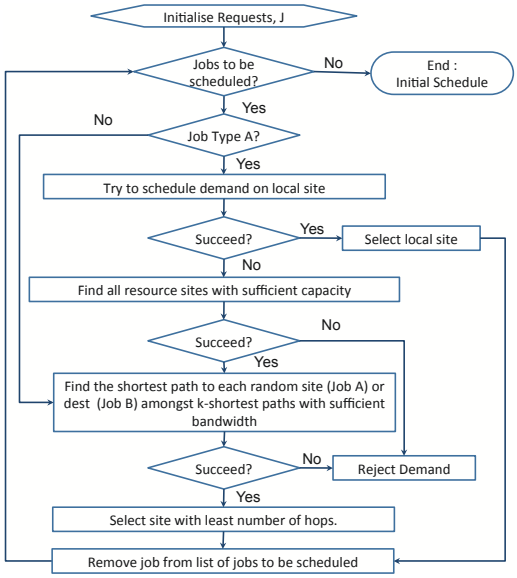


Fig. 3. Flowchart to Construct MSA Initial Solution

algorithm is shown in Algorithm 1. It starts off with an **initial solution** that has an initial **cost**. It then runs through a number of iterations at a given temperature, during which a new solution and its corresponding cost is obtained by a **perturbation** scheme. Depending on the cost, the solution from the perturbation is accepted or rejected.

The **cost** of a solution in the MSA is defined by the total number of hops used in satisfying a maximum number of requests. The flow-chart to create the **initial solution** is shown in Figure 3. For job-type A, the algorithm selects the closest IT resource site. Thus, it first attempts to use the IT resource site connected to the same edge node as the client node. If such an IT site exists and it has sufficient capacity, the job request is scheduled on that site. If such a site does not exist,

© 2011 ACADEMY PUBLISHER

the algorithm selects the closest IT resource site with sufficient resource capacity. It then routes the job request on the shortest path, $\kappa \in K, 1 \leq \kappa \leq K$, with sufficient bandwidth capacity. If a $(c, \kappa^{fwd}, \kappa^{rev})$ triple for $j^{A_{comp}}$ request or (s, κ^{fwd}) tuple for $j^{A_{stor}}$ request with sufficient capacity cannot be found, the request is rejected. This approach aims to create an initial solution using a minimum number of hops. For job-type B, the request is routed on the shortest path with sufficient bandwidth capacity among k-shortest paths, $\kappa \in K, 1 \leq \kappa \leq K$. If no such path is found, the request is rejected.

Algorithm 1 Modified Simulated Annealing

```
{J} set of requests to be scheduled
Initialize: T_min = 0.01, T_max = 300, T ← T_max
Initialize: cooling factor, c = 0.88
Initialize: currSoln ← Initial Solution using to flow-chart
set of unscheduled requests, availDemands ← J
while T > T_min do
    newSoln ← perturb currSoln
    Δ1 ← newSoln.cost1 − currSoln.cost1
    Generate u1 ∈ [0, 1] uniform random variable
    if Δ1 < 0 then
        currSoln ← newSoln
    else if Δ1 == 0 then
        Generate u2 ∈ [0, 1] uniform random variable
        Δ2 ← newSoln.cost2 − currSoln.cost2
        if Δ2 > 0 or u2 < P(Δ2) then
            currSoln ← newSoln
        end if
    else if u1 < P(Δ1) then
        currSoln ← newSoln
    end if
    T = T_max * c
end while
return Solution
```

The **perturbation** process generates new solutions in the neighbourhood of the current solution. It proceeds by freeing a random number of job requests and then attempting to re-orchestrate all unorchestrated job requests on a random IT resource site and a random path among the k-shortest paths. The change in cost, (Δ), between the new solution and the current solution is calculated. The solution is accepted or rejected according to the Boltzmann probability given in Equation (16).

$$P_{boltz}(\Delta) = \exp(\frac{-\Delta}{T}) \tag{16}$$

where P_{boltz} is the Boltzmann probability of accepting the new solution, T is a control parameter known as the current "temperature" and $-\Delta$ is the improvement to the solution.

E. Online Orchestration Algorithms

On-line algorithms are sequential in nature. They are based on the arrival time of job requests, with no knowledge of future jobs. That is, they consider job requests in a first-come first serve (FCFS) basis. Thus, in this scenario, a job is scheduled based on its requirements and the current state of the underlying resources. Two online algorithms, which are based on a novel cost-based algorithm are described and proposed. The Cost-Based Algorithm (CBA) attempts to orchestrate resources based on a *cost* it assigns to each potential resource configuration. It only considers resources with enough capacity to satisfy a request. It first creates a list of potential candidates, which is referred to as the Candidate List (CL) according to Algorithm 2. For Job-Type A that require IT resources, each potential destination, $r \in \mathcal{R}$ is considered. The k-shortest paths are calculated to each potential IT resource site using Dijkstra's algorithm on the infrastructure, G. If the path has sufficient resources, the *IT_site-path* pair, (r, κ) is stored in the CL. For Job-Type B requests, the destination is explicitly requested and Dijkstra's algorithm is implemented to the destination. The potential paths to the server node, (ϕ, κ) is stored in the CL. For generality, the pairs in the CL are referred to as the *resource-path* pair, $(\bar{d}, \kappa)$ for both Job-Type A and B, where $\bar{d}$ represents r or ϕ depending on if it refers to Job-Type A or Job-Type B.

Algorithm 2 CreateCandidateList

```
j current job to be scheduled
CL ← ∅
if Job-Type A then
    availResources ← get available resources r_v > r_j
end if
if Job-Type B then
    availResources ← φ
end if
for all d̄ ∈ availResources do
    Calculate k paths to resource d̄
    for all (d̄, κ) pair do
        CL ← CL ∪ (d̄, κ)
    end for
end for
return CL
```

The cost φ, applicable to each possible *destination-path* pair, for each job, $\varphi_{\bar{d},\kappa}$ is calculated. The resource configuration with the minimum cost is selected. The following notation and parameter definitions are appropriate for the calculation of each of the costs for the CBA:

- Let $b_{\bar{d},k}$ be the bandwidth of the path. Let $b_{i,\bar{d},k}$ be the available bandwidth on link i, which belongs to path k leading to destination $\bar{d}$, $k = 1, 2, 3...K$, where k represents the k-th path. $b_{\bar{d},k} = \min_{i \in I}\{b_{i,\bar{d},k}\}$. Among all k-paths, $b_{max} = \max_{k \in K}\{b_{i,\bar{d},\kappa}\}$.
- Let r_m be the available resource capacity on a resource site, m. $r_{max} = \max_{m \in M}\{r_m\}$.
- Let $h_{\bar{d},\kappa}$ be the number of hops of path κ to destination $\bar{d}$. Among all k-paths, $h_{max} = \max_{k \in K}\{h_{\bar{d},\kappa}\}$.

The costs that are defined for the on-line CBA algorithm are:

© 2011 ACADEMY PUBLISHER

1) Online Resource Usage Resource Orchestration Algorithm, ORU: This algorithm selects the closest IT resource site with sufficient capacity over the bandwidth constrained shortest path. The logic behind this approach is to minimize the number of network nodes and links used in satisfying a job request. Thus fulfilling the objective function of the ILP. The bandwidth constrained shortest path first algorithm is used to select the path to the IT resource site, $r \in R$, or content server $\phi \in \Phi$. The cost is defined as follows:

$$\varphi_{\bar{d},\kappa} = \frac{h_{\bar{d},\kappa}}{h_{max}} \tag{17}$$

It differs from the closest resource site algorithm (CRS) presented in [46] in that it takes the residual bandwidth capacity of the links in the underlying network infrastructure in selecting IT resource sites.

2) Adaptive Service Aware Resource Orchestration Algorithm, ASARO: This cost jointly takes into account network and IT resources to compose network-IT services that satisfy job requirements. The algorithm uses a cost that is based on the utilisation levels of both network links and IT resource sites to determine which site to select and which of the k-shortest path to use to reach the selected IT resource site. In addition, it introduces weight coefficients, α and β, which are selected based on the utilisation of the network and IT resources respectively such that the selection of a *destination-path* tuple is adapted to the state of the underlying infrastructure. To elaborate, if there are two resource sites with sufficient capacity, each with paths of equivalent hop distance, the α and β values will be the determining factor that decides which *destination-path* pair will be selected. If the network bandwidth resources are more utilised, i.e. if there is a network bandwidth bottleneck, then the α coefficient is greater than the β coefficient such that the minimisation of the cost returns a solution that results in lower overall network utilisation. Likewise for the IT resources. If there is a scarcity of IT resources, then the value of β is greater than α. The algorithm first computes the average link bandwidth and IT resource usage. It then finds the constrained shortest path to each IT resource site that has sufficient resource capacity. The constrained shortest path first algorithm first prunes all links that do not meet the bandwidth requirement of a new request. Then the shortest path computation is run on the pruned topology. For each returned result, the cost function is calculated according to Equation (18).

The cost is calculated as follows:

$$\varphi_{\bar{d},\kappa} = \frac{h_{j,\bar{d},\kappa}}{h_{max}} \times \left(\alpha \cdot \frac{b_{max}}{b_{\bar{d},\kappa}} + \beta \cdot \frac{r_{max}}{r_m} \right) \tag{18}$$

The cost function ensures that jobs which require IT resources are assigned to the closest IT resource site with the most-free resource capacity. Jobs that do not require IT resources are assigned to resources with a trade-off between hop distance and available link bandwidth capacity. The solution with the minimum cost, $\varphi'_{\bar{d},\kappa}$, is selected as the resource configuration on which to schedule the job, j. If there is not enough resources to schedule the request, it is rejected.

The minimisation function for the cost is defined as:

$$\varphi'_{\bar{d},k} = f(k) = \begin{cases} \varphi_{\bar{d},\kappa} & \text{if } k = 0 \\ \min\left(f(k-1), \varphi_{\bar{d},k}\right) & \text{if } 1 \leq k \leq K \end{cases} \tag{19}$$

where

$$\bar{d} = \begin{cases} r & \text{if Job-Type A} \\ \phi & \text{if Job-Type B} \end{cases}$$

V. Validation and Analysis

For our analytical model and simulation, we consider a 5-node topology to evaluate our ILP and to validate the heuristic solution. We then do a scalable analysis using the NSFNET and the European Optical Network (EON). Finally, the online algorithms are then studied, again over the NSFNET and EON.

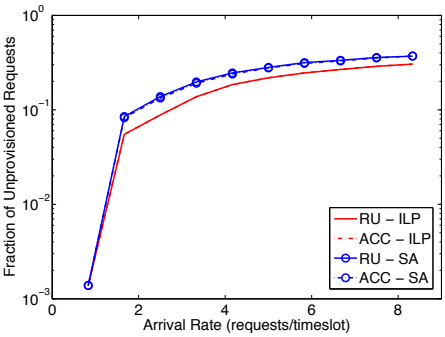
A. Evaluation of Mathematical Approach and Validation of the Heuristic Solution

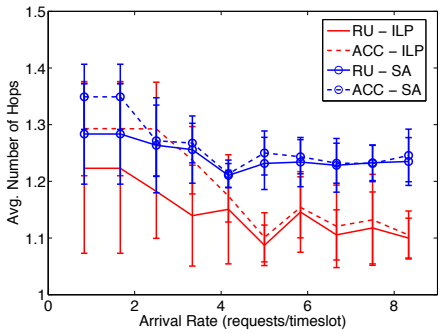
We first evaluate our ILP and use it to validate the heuristic over a 5-node topology connecting four resource nodes for tractability. The link bandwidth is set to 40 Gbps, with 8 wavelengths per link. The resource capacity of each storage/computational node is randomly selected from $\{1250, 1500\}$. The request set used in this analysis is created randomly. We recall that Job Type A requests are storage or computational requests and Job Type B requests are video requests. The start time is uniformly distributed between $[0, 22]$ for Job Type B requests and $[0, 15]$ for Job Type A requests. The duration of each job request is uniformly distributed between $[2, 5]$ for Job Type B and $[5, 23]$ for Job-Type A requests. Request for storage/computational resources is uniformly distributed between $[10, 50]$. Bandwidth requirements are uniformly distributed between $[50, 100]$ Gbps for Job Type B and $[100, 150]$ Gbps for Job Type A. The granularity of each wavelength is 1 Gbps. Thus bandwidth can only be assigned in discrete quantities of 1 Gbps. The number of alternate paths is set to $K = 3$. We vary the number of job requests we simulate between 20 and 200 requests, over 24 timeslots and the results are averaged over 20 simulation runs for which we calculate a 95% confidence interval. We use an Intel Core $i7$, 2.93 GHz machine, with 8 GB of RAM.

Fig. 4 shows the ILP and heuristic solution for both the RU and the ACC objectives. Recall that RU is our proposed ILP and ACC is an ILP whose objective is to maximise the number of accepted requests [43]. To implement ACC, we take into consideration only relevant constraints from [43] and extend it to introduce video streaming, storage requests, the corresponding resources and relevant constraints.

We first measure the performance of our proposed service-oriented resource orchestration ILP, RU-ILP, by comparing it to the ACC-ILP. Fig.4(a) shows the fraction of unprovisioned requests measured against the arrival rate which is the number of requests per timeslot. We observe that RU returns the same results when compared to ACC in terms of rejection ratio. Fig. 4(b) shows that RU outperforms ACC in terms of average

© 2011 ACADEMY PUBLISHER





(b) Number of Hops

Fig. 4. Comparing two Objective Functions over the 5-Node Topology

number of hops, using 10% less hops on average than ACC, with the least hops used at lower arrival rates.

Next, we validate the proposed heuristic, RU-SA, by comparing it to the ILP formulation, RU-ILP which returns optimal solutions. We observe that our heuristic solution is comparable with the optimal ILP solution in terms of the fraction of unprovisioned requests and number of hops with a 5% difference at the highest arrival rate in both cases.

B. Evaluation of the Offline Heuristics

Finally, we evaluate the performance of our proposed resource orchestration heuristic, RU against two other heuristics as benchmarks: (i) the ACC heuristic, which was validated against the ACC-ILP in Fig. 4 and (ii) an approach which assumes no cross-domain interactions and no cooperation between InPs. We call this the NoSP. In the NoSP, a request is made to each InP one at a time until the complete end-to-end service can be achieved. For scalability, we perform the evaluation over the 32 node EON with 92 unidirectional links, 9 storage resource sites and 9 computational sites and the NSFNET with 14 nodes, 42 unidirectional links, 5 storage resource sites and 5 computational sites. Each link has a bandwidth of 40 Gbps and 16 wavelengths. The storage and computational capacities of each edge nodes is chosen randomly from $[2000, 7000]$ units with a step size of 1000 units. The start time is uniformly distributed between $[0, X-5]$ for Job Type B requests and $[0, X-10]$ for Job Type A requests. X represents the maximum number of timeslots which ranges from $[20, 100]$. The number of requests during these timeslots is 2000, thus the arrival rate (requests/timeslot) is varied. The duration of each job request is uniformly distributed between $[2, 5]$ for Job Type B and $[4, 12]$ for Job-Type A requests. Request for storage/computational resources is uniformly distributed between $[50, 200]$. Bandwidth requirements are uniformly distributed between $[2, 5]$ Gbps for video requests and $[10, 50]$ Gbps for storage/computational requests. The source and the destination for video requests are randomly selected.

We first compare RU and ACC over the larger NSF and EON topologies. Fig. 5 and 6 show that on comparison, ACC and RU give the same solution in terms of fraction of unprovisioned requests, but RU outperforms ACC in terms of average number of hops and average resource usage. We observe an improvement in network usage, but negligible improvement in IT resource usage. This is due to the fact that RU uses less links on average, thus uses the network resources more efficiently than ACC.

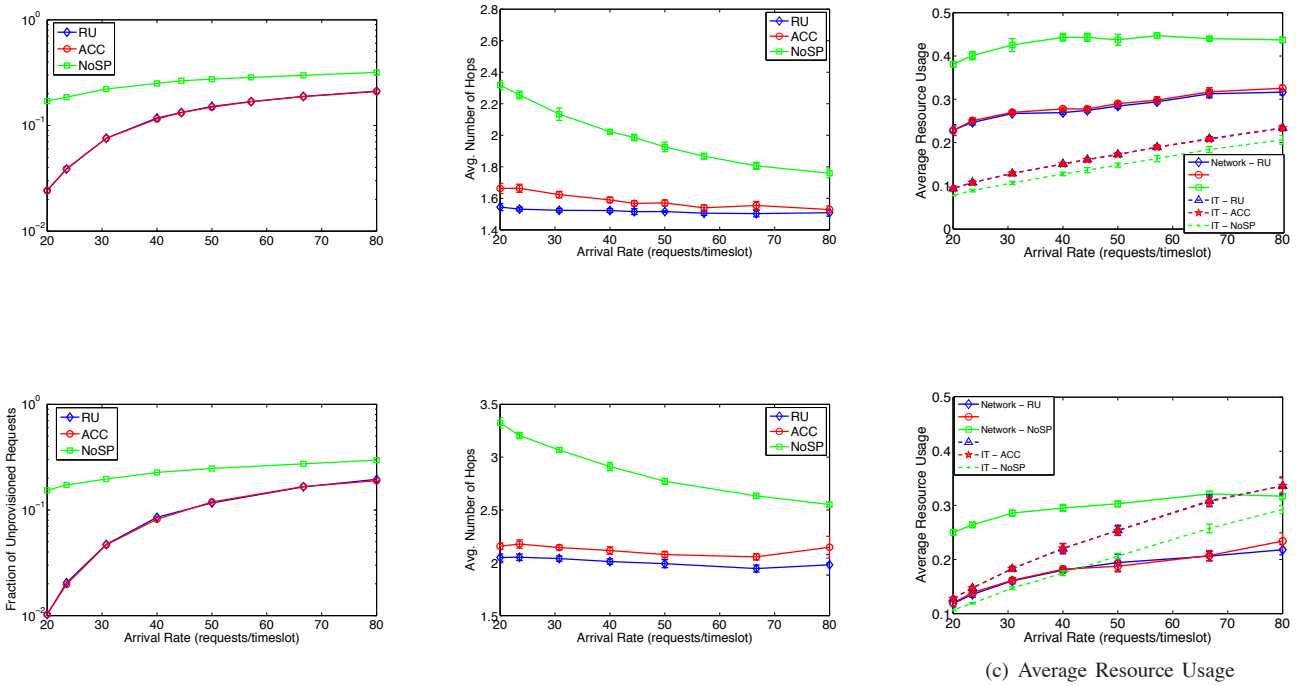
Finally, we compare our resource orchestration (RU) heuristic with the NoSP approach where there is no cooperation between InPs. For the fraction of unprovisioned requests shown in Fig. 5(a) and 6(a), we observe that when resource orchestration is implemented over the EON and NSFNET as opposed to the NoSP, there is 14% and 15% more accepted requests respectively on average. There is also an improvement of 24% and 30% in the average number of hops used on average over the NSFNET (Fig. 5(b)) and EON (Fig. 6(b)) respectively when the RU is used as opposed to the NoSP. From Fig. 5(c) and 6(c), we observe an average improvement of 37% in the network resource usage and 17% improvement in the IT resource usage on the NSFNET. For the EON, we observe an improved usage of 22% and 41% for the network and IT resources respectively. From our result, we observe that our RU approach performs much better in terms of fraction of unprovisioned requests, number of hops and average resource usage than the NoSP approach. This can be attributed to the collaboration that exists when using the resource orchestration of the SP. As the SP is able to globally optimise the selection of resources from amongst the multiple InPs that may be able to satisfy a users requirement in such a way that the resources are optimally utilised, it results in an increase in the number of accepted requests.

C. Evaluation of the Online Algorithms

The online algorithms are evaluated over the NSFNET with 14 nodes, 42 unidirectional links, 5 storage resource sites and 5 computational sites, and the 32 node EON with 92 unidirectional links, 9 storage resource sites and 9 computational sites. 50% of the links have a bandwidth capacity of 2.5 Gbps, and the other 50% have a bandwidth capacity of 10 Gb/s. Each link has 8 wavelengths. The storage and computational capacities of each edge node is chosen randomly from $[150, 300]$ and $[3000, 8000]$ units respectively.

Requests arrive to the network according to a Poisson process with an exponentially distributed inter-arrival time and

© 2011 ACADEMY PUBLISHER



(c) Average Resource Usage

Fig. 6. Evaluation of Resource Orchestration over the EON Topology

TABLE I
REQUEST INFORMATION

	Video1	Video2	Storage	Compute
Bandwidth (Mbps)	$\{20, 900\}$	$[2, 10]$	$[1000, 5000]$	$[1000, 5000]$
Duration (hours)	3	2	$5/[4, 6]$ [1]	7
Subscribers	10	100	1	1
Storage (TB)			$[1, 20]$	
MIPS				$[20, 100]$
Storage Devices				$\lceil 1, 3 \rceil$
Processors			$\lceil 1, 4 \rceil$	

holding time (duration) as tabulated in Table I. The processing and storage request parameters as well as the bandwidth are uniformly distributed between the ranges shown in Table I. The source and destination pairs are selected randomly from among the edge nodes. The simulation was run for $20,000$ requests. The first $5,000$ requests are ignored in the performance metric to introduce activity in the underlying infrastructure before computing the metrics.

To evaluate the two online algorithms, ORU, and ASARO, they are compared against three benchmark algorithms: The legacy, closest resource site (CRS) and the constrained crankback (CCB) algorithms.

The legacy algorithm is analogous to the traditional 2-step resource scheduling algorithm as it takes place in 2 steps. For Job-Type A, the algorithm first selects an IT site which it then presents as a destination to the network control plane as the destination for a connection request. For Job-Type B, since the destination is already known, a connection request is made directly to the network control plane. This algorithm is referred to as "2-Step" since the selection of a path and a resource site is addressed independent of each other. Furthermore, it does not take into account the status of the network link capacities in selecting a suitable IT resource site. It assumes that there will be sufficient bandwidth on the network links to the selected IT resource site.

The closest resource site algorithm improves on the Legacy algorithm. The main difference between the two algorithms, is that the algorithm first attempts to schedule the job on local resources, and if this is infeasible it tries to submit the request to a resource site that is as close as possible to the requesting node. Thus it minimises the amount of network links that data has to be sent over.

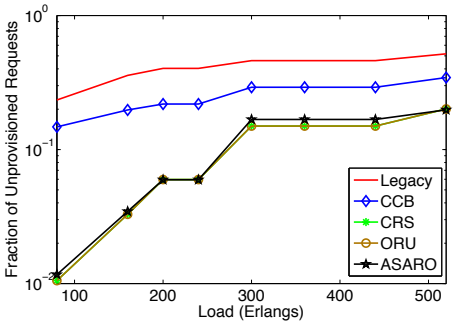
Unlike the ORU algorithm, the CRS algorithm does not take into account bandwidth levels when selecting the IT resource site. Like the Legacy algorithm, it assumes that there will be sufficient bandwidth on the network links to the selected IT resource site.

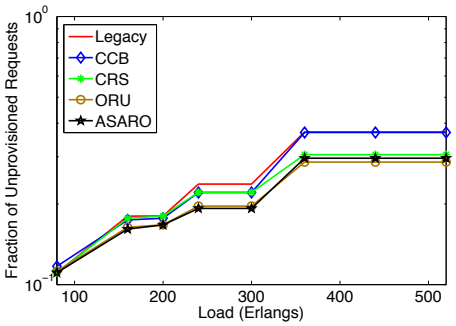
The constrained crank-back algorithm improves on the legacy algorithm. It is also a benchmark algorithm. It provides a feed-back mechanism that allows Job-Type A requests to be cranked-back if no network path with sufficient bandwidth to the selected IT resource site resource can be found. In this way, a different IT resource site can be selected until a network path with sufficient bandwidth resource can be found.

If the job request is of type A, the IT resource sites are arranged randomly. Then the job is assigned to the first random IT resource site that can satisfy the job's requirement. The selected site is the destination of the request. If the job is of type B, this step is omitted, since the destination is explicitly stated

[1]Data transfer time is according to an exponential distribution with a mean of 5hours and stored between [4,6] hours according to a uniform distribution

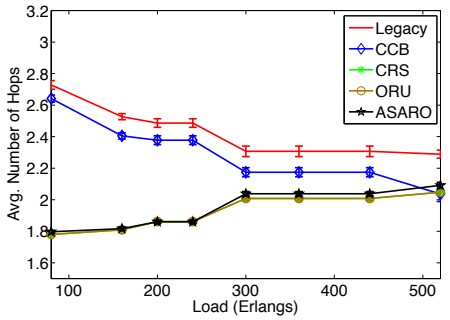
© 2011 ACADEMY PUBLISHER



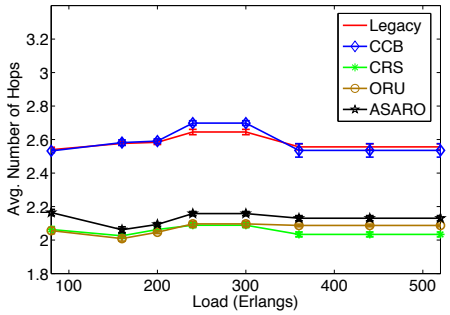


(b) EON

Fig. 7. Evaluating the Fraction of Unprovisioned Requests of Online Resource Orchestration Algorithms over the (a) NSFNET and (b) EON



(a) NSFNET



(b) EON

Fig. 8. Evaluating the Average Number of Hops of the Online Resource Orchestration Algorithms over the (a) NSFNET and (b) EON

in the job request. The algorithm then proceeds to attempt to find a path with sufficient bandwidth to the destination. If no path is found, the request is crank-backed. The next IT resource site in the randomly ordered set is considered. The algorithm repeats itself until an IT resource site that can satisfy the storage/computational resource requirement is selected or until all feasible *destination-path* pair are exhausted. If the network is not congested, the constraint crankback algorithm behaves like the legacy algorithm, since there will be no need for the crank-back mechanism.

We recall that the ORU algorithm is similar to ILP model formulation presented in this paper and thus has the same objective as the RU heuristic. Like the RU heuristic, the ORU attempts to find resource-path combinations that minimise the resource usage in terms of number of hops.

Figure 7 shows that the three proposed algorithms perform much better than the benchmark algorithms in terms of fraction of unprovisioned requests. From the figure, its is observed that amongst the three benchmark algorithms, the CRS algorithm has the best results. This is because the CRS algorithm is able to schedule requests on the user site, thus reducing the chance of blocking due to bandwidth limitations. The CCB performs better than Legacy. This is because of the crank-back process in the CCB algorithm, where the algorithm is repeated across all possible *resource-path* pair until a suitable solution found.

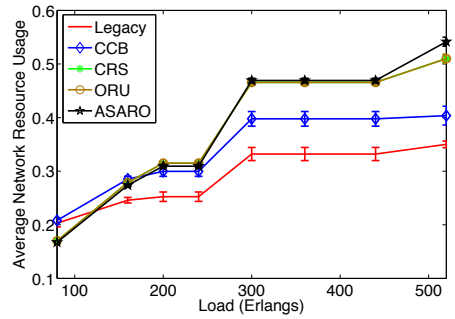
The two proposed algorithms: the ORU and ASARO holistically view the resources from multiple InPs, stored in the SP marketplace. Thus are able to make better choices regarding which *resource-path* pair to select and from which InP. As seen from Figure 7, implementing the ORU results in a 8% and 38% improvement on average when compared to the CCB and Legacy algorithms respectively over the NSFNET. However, there is a negligible difference when compared to the CRS algorithm in the NSFNET. This difference is more prominent in the EON, performing consistently better than the CRS algorithm over all the loads.

The results also show that there is little gain in considering a trade-off between hop distance, free bandwidth and IT resource capacity, as implemented by the ASARO algorithm. This algorithms perform negligibly better than the ORU algorithm (which performs better than the benchmark algorithms) in terms of success ratio for both the NSFNET and EON. In the EON, at high loads, the ORU outperforms the ASARO algorithm.

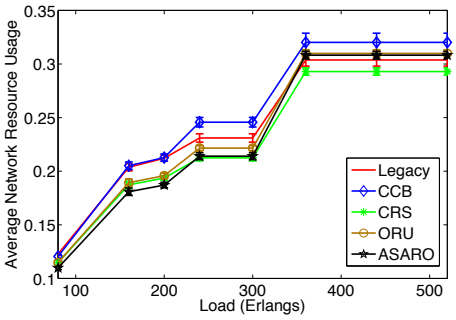
As expected, increasing the load, leads to a reduction in the number of accepted requests.

Figure 8 measures the average number of hops used by each of the online algorithms and the benchmark algorithms. The Legacy and CCB algorithms are the least efficient, using 27% (0.5 hops) and 21% (0.4 hops) more hops on average in the NSFNET respectively and 24% (0.5 hops) and 25% (0.5 hops) more hops on average in the EON respectively than the other algorithms. This is expected since there is little or no co-operation between the different providers involved in composing services to satisfy user requests. Of the three benchmark algorithms, the CRS algorithm is the most opti-

© 2011 ACADEMY PUBLISHER

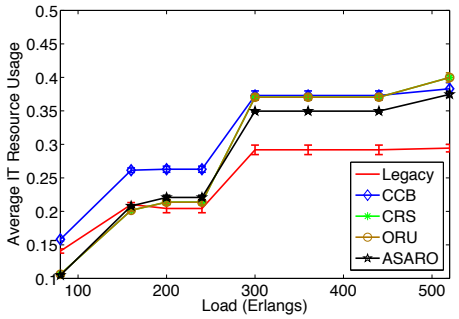


(a) NSFNET

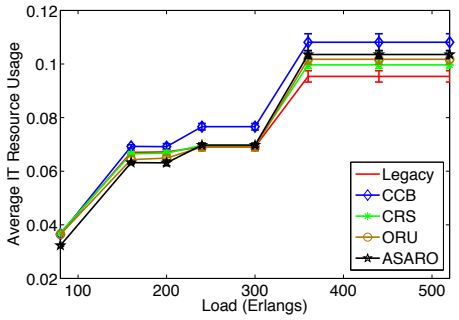


(b) EON

Fig. 9. Evaluating the Average Network Resource Usage of the Online Resource Orchestration Algorithms over the (a) NSFNET and (b) EON



(a) NSFNET



(b) EON

Fig. 10. Evaluating the Average IT Resource Usage of the Online Resource Orchestration Algorithms over the (a) NSFNET and (b) EON

mised since it always uses the closest IT resource site, hence performs comparatively to the proposed resource orchestration algorithms. The two 'least hop' algorithms (ORU and CRS) give the best results in terms of average number of hops.

The average network and IT resource usage for the three algorithms were measured and illustrated in Figures 9 and 10. As expected, increasing the load leads to an increase in average resource usage. In the NSFNET, the benchmark algorithms have less network resource usage due to the higher blocking they exhibit. However, the CRS has a high average network usage due to the higher acceptance ratios. In the EON, due to the average node degree and the size of the network, the $k > 1$ paths are much longer than the $k = 1$ path (as opposed to the NSFNET, which gives similar results for $k = 2$ and $k = 1$) thus, the benchmark algorithms, which use the longer paths (Figure 8), use higher amounts of network resources.

Finally, Figure 10 shows that the Legacy algorithm has the lowest IT resource usage, and the CCB has the highest IT resource usage. This result is trivial since the selection of IT resource sites is random when using the Legacy algorithm, which has high levels of unprovisioned requests. The CCB on the other hand iteratively tries to find a *resource-path* pair with sufficient capacity, hence ends up using the least efficient *resource-path* pairs in terms of IT and network resource usage. The proposed algorithms exhibit similar results in terms of IT resource usage.

VI. Conclusion

Emerging applications are motivating the need for more service oriented architectures. In this paper, we discussed the peculiarities introduced by these applications which need to be addressed. We introduced a service oriented framework, SOAFI as a possible solution to address these peculiarities, particularly as a means to organise resources from multiple domains. Central to this framework is a service oriented resource orchestration mechanism that coordinates resources from different InPs, facilitated by a unified platform, the Service Plane. We developed a new mathematical model for the service-oriented resource orchestration problem and presented a heuristic approximation based on a modified Simulated Annealing heuristic for efficiency and scalability. Our results show that our proposal is comparable to previous studies related to co-scheduling of resource. Moreover our solution outperforms these in terms of resource usage. Our ILP, which returns optimal results, is used to benchmark our heuristic approximation. A scalability analysis of our heuristic approximation is carried out by considering the European Optical Network and the NSFNET to satisfy a larger number of requests. Finally, two algorithms: ORU and ASARO were proposed for operational on-demand scenarios and compared against three benchmark algorithms used in literature. The proposed resource orchestration algorithms outperformed the three benchmark algorithms. The results shows that the proposed SP resource orchestration performs much better than when there is no orchestration of resources amongst multiple

© 2011 ACADEMY PUBLISHER

InPs.

ACKNOWLEDGMENT

The work described in this paper was carried out with the support of the GEYSERS (Generalized Architecture for Dynamic Infrastructure Services), and the Mantychore projects funded by the European Commission through the 7th ICT Framework Programme.

REFERENCES

[1] F. Callegati, A. Campi, G. Corazza, D. Simeonidou, G. Zervas, Y. Qin, and R. Nejabati, "SIP-empowered optical networks for future IT services and applications," *Communications Magazine, IEEE*, vol. 47, no. 5, pp. 48 –54, May 2009.

[2] M. Maeda and K. Thompson, "Cyberinfrastructure and large-scale network testbeds," in *Optical Fiber Communication Conference, 2004. OFC 2004*, vol. 2, 2004, p. 1 pp. vol.2.

[3] D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, A. Tzanakaki, and M. O'Mahony, "Dynamic optical-network architectures and technologies for existing and emerging Grid services," *Lightwave Technology, Journal of*, vol. 23, no. 10, pp. 3347 – 3357, 2005.

[4] D. Simeonidou, R. Nejabati, M. J. OMahony, A. Tzanakaki, and I. Tomkos, "An optical network infrastructure suitable for global Grid computing," in *TERENA Networking Conference 2004*, June 2004.

[5] I. Gorton, P. Greenfield, A. Szalay, and R. Williams, "Data-intensive computing in the 21_{st} century," *Computer*, vol. 41, pp. 30–32, April 2008.

[6] G. Bell, J. Gray, and A. Szalay, "Petascale computational systems," *Computer*, vol. 39, no. 1, pp. 110 – 112, 2006.

[7] C. Bozzi, T. Adye, D. Andreotti, E. Antonioli, R. Barlow, B. Bense, D. Boutigny, C. Brew, D. Colling, R. Cowles, P. Elmer, E. Feltresi, A. Forti, G. Grosdidier, A. Hasan, H. Lacker, E. Luppi, J. Martyniak, A. McNab, A. Petzold, D. Smith, J. Sundermann, and P. Veronesi, "Using the Grid for the BaBar experiment," *Nuclear Science, IEEE Transactions on*, vol. 51, no. 5, pp. 2045 – 2049, 2004.

[8] C. Bozzi, "Using Grid for the BaBar experiment," in *Nuclear Science Symposium Conference Record, 2003 IEEE*, vol. 3, 2003, pp. 1626 – 1629 Vol.3.

[9] H. B. Newman, M. H. Ellisman, and J. A. Orcutt, "Data-intensive e-science frontier research," *Commun. ACM*, vol. 46, pp. 68–77, November 2003.

[10] G. Zervas, V. Martini, Y. Qin, E. Escalona, R. Nejabati, D. Simeonidou, F. Baroncelli, B. Martini, K. Torkmen, and P. Castoldi, "Service-oriented multigranular optical network architecture for clouds," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 10, no. 2, pp. 883 –891, 2010.

[11] G. Zervas, M. De Leenheer, L. Sadeghioon, D. Klonidis, Y. Qin, R. Nejabati, D. Simeonidou, C. Develder, B. Dhoedt, and P. Demeester, "Multi-granular optical cross-connect: Design, analysis, and demonstration," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 1, no. 1, pp. 69 –84, 2009.

[12] G. Zervas, Y. Qin, R. Nejabati, D. Simeonidou, F. Callegati, A. Campi, and W. Cerroni, "SIP-enabled optical burst switching architectures and protocols for application-aware optical networks," *Computer Networks*, vol. 52, no. 10, pp. 2065 – 2076, 2008, challenges and Opportunities in Advanced Optical Networking.

[13] G. Zervas, E. Escalona, R. Nejabati, D. Simeonidou, G. Carrozzo, N. Ciulli, B. Belter, A. Binczewski, M. Poznan, A. Tzanakaki, and G. Markidis, "Phosphorus grid-enabled GMPLS control plane (GM-PLS): architectures, services, and interfaces," *Communications Magazine, IEEE*, vol. 46, no. 6, pp. 128 –137, june 2008.

[14] N. Ciulli, G. Carrozzo, G. Giorgi, G. Zervas, E. Escalona, Y. Qin, R. Nejabati, D. Simeonidou, F. Callegati, A. Campi, W. Cerroni, B. Belter, A. Binczewski, M. Stroinski, A. Tzanakaki, and G. Markidis, "Architectural approaches for the integration of the service plane and control plane in optical networks," *Optical Switching and Networking*, vol. 5, no. 2-3, pp. 94 – 106, 2008, advances in IP-Optical Networking for IP Quad-play Traffic and Services.

[15] A. Jukan and G. Karmous-Edwards, "Optical control plane for the Grid community," *Communications Surveys Tutorials, IEEE*, vol. 9, no. 3, pp. 30 –44, 2007.

[16] G. Kamous-Edwards and A. Jukan, "An optical control plane for the Grid community: opportunities, challenges, and vision," *Communications Magazine, IEEE*, vol. 44, no. 3, pp. 62 – 63, 2006.

[17] T. Sekiguchi, K. Ono, A. Kotabe, and Y. Yanagawa, "1-Tbyte/s 1-Gbit DRAM architecture using 3-D interconnect for high-throughput computing," *Solid-State Circuits, IEEE Journal of*, vol. PP, no. 99, p. 1, 2011.

[18] L. Yan, B. Wu, Y. Wen, S. Zhang, and T. Chen, "A reconfigurable processor architecture combining multi-core and reconfigurable processing unit," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 292010-july1 2010, pp. 2897 –2902.

[19] Y. Qi, Z. Zhou, Y. Wu, Y. Xue, and J. Li, "Towards high-performance pattern matching on multi-core network processing platforms," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, 2010, pp. 1 –5.

[20] C. Zhong, Z.-Y. Qu, F. Yang, and M.-X. Yin, "Parallel multisets sorting using aperiodic multi-round distribution strategy on heterogeneous multi-core clusters," in *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on*, 2010, pp. 247 –254.

[21] T. Jepson, "The basics of reliable distributed storage networks," *IT Professional*, vol. 6, no. 3, pp. 18 – 24, 2004.

[22] H.-Y. Lin and W.-G. Tzeng, "A secure decentralized erasure code for distributed networked storage," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 11, pp. 1586 –1594, 2010.

[23] Y. Tanimura, K. Hidetaka, T. Kudoh, I. Kojima, and Y. Tanaka, "A distributed storage system allowing application users to reserve I/O performance in advance for achieving SLA," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, 2010, pp. 193 –200.

[24] L. Costa and M. Ripeanu, "Towards automating the configuration of a distributed storage system," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, 2010, pp. 201 –208.

[25] D. Leong, A. G. Dimakis, and T. Ho, "Distributed storage allocation for high reliability," in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1 –6.

[26] X.-Y. Yang, Z. Liu, W. Zhang, and D.-T. Guo, "A high-efficiency data distribution algorithm in distributed storage," in *Information Assurance and Security, 2009. IAS '09. Fifth International Conference on*, vol. 1, 2009, pp. 627 –630.

[27] D. Simeonidou, R. Nejabati, B. S. Arnaud, M. Beck, P. Clarke, D. B. Hoang, D. Hutchison, G. Karmous-Edwards, T. Lavian, J. Leigh, J. Mambretti, V. Sander, J. Strand, and F. Travostino, *Optical Network Infrastructure for Grid*, Global Grid Forum Std. GFD-I.036, October 2004.

[28] M. J. O'Mahony, C. Politi, D. Klonidis, R. Nejabati, and D. Simeonidou, "Future optical networks," *Lightwave Technology, Journal of*, vol. 24, no. 12, pp. 4684 –4696, dec. 2006.

[29] I. Rodero, F. Guim, and J. Corbalan, "Evaluation of coordinated Grid scheduling strategies," in *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, 2009, pp. 1 –10.

[30] H.-J. Choi, E. Kim, Y. Lee, H. Yeom, D. Nam, and S. Hwang, "A super-metascheduler-based approach for integrating multiple heterogeneous Grids," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 03, 2009, pp. 2065 –2070.

[31] V. Garonne, A. Tsaregorodtsev, and E. Caron, "A study of meta-scheduling architectures for high throughput computing: Pull versus push," in *Parallel and Distributed Computing, 2005. ISPDC 2005. The 4th International Symposium on*, 2005, pp. 226 –233.

[32] (2006, October) Reference model for service oriented architecture 1.0. OASIS Standard. [Online]. Available: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html

[33] C. Abosi, R. Nejabati, and D. Simeonidou, "Service oriented resource orchestration in future optical networks," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, 2011, pp. 1 –6.

[34] ——, "A novel service composition mechanism for future optical internet," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 1, pp. A106–A120, July 2009.

[35] ——, "A service plane architecture for future optical internet," in *Optical Network Design and Modeling, 2009. ONDM 2009. International Conference on*, 2009, pp. 1 –6.

[36] ——, "Design and development of a semantic information modelling

© 2011 ACADEMY PUBLISHER

framework for a service oriented optical internet," *Journal of Networks*, vol. 5, no. 11, 2010.
[37] Z. Sun, W. Guo, Z. Wang, Y. Jin, W. Sun, W. Hu, and C. Qiao, "Scheduling algorithm for workflow-based applications in optical grid," *J. Lightwave Technol.*, vol. 26, no. 17, pp. 3011–3020, 2008. [Online]. Available: http://jlt.osa.org/abstract.cfm?URI=JLT-26-17-3011
[38] X. Liu, C. Qiao, W. Wei, X. Yu, T. Wang, W. Hu, W. Guo, and M.-Y. Wu, "Task scheduling and lightpath establishment in optical grids," *J. Lightwave Technol.*, vol. 27, no. 12, pp. 1796–1805, 2009. [Online]. Available: http://jlt.osa.org/abstract.cfm?URI=JLT-27-12-1796
[39] R. McClatchey, A. Anjum, H. Stockinger, A. Ali, I. Willers, and M. Thomas, "Data intensive and network aware (diana) grid scheduling," *Journal of Grid Computing*, vol. 5, pp. 43–64, 2007, 10.1007/s10723-006-9059-z.
[40] W. Guo, W. Sun, Y. Jin, W. Hu, and C. Qiao, "Demonstration of joint resource scheduling in an optical network integrated computing environment [topics in optical communications]," *Communications Magazine, IEEE*, vol. 48, no. 5, pp. 76 –83, may. 2010.
[41] V. T. De, B. Volckaert, P. Thysebaert, M. D. Leenheer, F. D. Turck, B. Dhoedt, and P. Demeester, "Network aware scheduling in grids," in *NOC '04: Proc. 9th European Conference on Networks and Optical Communications*, Eindhoven, The Netherlands, 2004.
[42] R. Aoun, M. Gagnaire, "Service differentiation based on flexible time constraints in market-oriented grids," accepted for publication at GLOBECOM. HNL, USA (2009).
[43] H.-H. Nguyen, M. Gurusamy, and L. Zhou, "Provisioning lightpaths and computing resources for location-transparent scheduled grid demands," in *Optical Network Design and Modeling, 2008. ONDM 2008. International Conference on*. IEEE, 12-14 2008, pp. 1 –6.
[44] Y. Wang, Y. Jin, W. Guo, W. Sun, W. Hu, and M.-Y. Wu, "Joint scheduling for optical grid applications," *J. Opt. Netw.*, vol. 6, no. 3, pp. 304–318, Mar 2007.
[45] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953.
[46] M. De Leenheer, C. Develder, T. Stevens, B. Dhoedt, M. Pickavet, and P. Demeester, "Design and control of optical grid networks," in *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, sept. 2007, pp. 107 –115.

Chinwe E Abosi received her PhD from the Electronic Systems Engineering Department of the University of Essex, United Kingdom in 2011. She received her MS degree in Information Networking from Carnegie Mellon University, Pittsburgh, Pennsylvania, in association with the Athens Information Technology Centre, Athens, Greece in 2006 and her BEng degree in Electrical and Electronics Engineering from the University of Botswana in 2005.

Reza Nejabati joined University of Essex in 2002 and he is currently a member of Photonic Network Group at the University of Essex. During the last 11 years he has worked on ultra high-speed optical networks, service oriented and application-aware networks, network service virtualization, control and management of optical networks, high-performance network architecture and technologies for e-science. Reza Nejabati holds a PhD in optical networks and an MSc with distinction in telecommunication and information systems from University of Essex, Colchester, United Kingdom.

Dimitra Simeonidou is currently a professor at the University of Essex. She has over 15 years experience in the field of optical transmission and optical networks. In 1987 and 1989 she received a BSc and MSc from the Physics Department of the Aristotle University of Thessalonica, Greece and in 1994 a PhD degree from the University of Essex.

From 1992 to 1994 she was employed as Senior Research Officer at University of Essex in association with the MWTN RACE project. In 1994 she joined Alcatel Submarine Networks as a Principle Engineer and contributed to the introduction of WDM technologies in submerged photonic networks. She participated in standardisation committees and was an advising member of the Alcatel Submarine networks patent committee.

Professor Simeonidou is the author over 350 papers and the holds 11 patents relating to photonic technologies and networks. Main research interests include optical wavelength and packet switched networks, network control and management and GRID networking.

© 2011 ACADEMY PUBLISHER