Mean Field Calculation for Optical Grid Dimensioning

Benny Van Houdt, Chris Develder, Juan F. Pérez, Mario Pickavet, Bart Dhoedt

***Abstract*—For traditional optical network dimensioning, a plethora of algorithms exists to design the amount of network resources required to accommodate a given amount of traffic, expressed as a (source,destination)-based traffic matrix. In optical Grid dimensioning however, the anycast principle applies: Grid users do not really care where exactly their tasks (Grid jobs) end up being executed. Thus, the destination of traffic is not known beforehand and traditional dimensioning algorithms are not applicable. In this paper we propose a mean field calculation method to analytically derive the traffic matrix for given job arrival intensities at the originating Grid sites (the sources). We also indicate how it can be integrated in a step-wise dimensioning approach to compute not only the amount of network resources, but also Grid resources (computational and/or storage). Hence it forms part of a solution for Grid dimensioning: determining how many servers to provide, where to place them, and which network to install for interconnecting server sites and users generating Grid jobs.**

***Index Terms*—Grids, Dimensioning, Mean Field Calculation**

I. Introduction

In several research fields, the need arose to build powerful computer systems to face computational and data storage challenges (e.g. particle physics, astrophysics, etc.). To meet the demand for a huge common resource pool to process the tasks (jobs) at hand, networks interconnecting cluster centers were deployed. This led to the creation of so-called Grids. More recently, the potential of Grid infrastructure for more consumer/business oriented applications was acknowledged by industry, and referred to as cloud computing [1]. (In this paper, we will stick to the term Grids to also include cloud computing.) To realize the interconnecting Grid network, optical technology is the solution of choice, able to meet both the high data rates typical of many e-science applications and the low latency requirements associated with most business/consumer solutions. Grids based on optical network infrastructure promise to offer cost and resource efficient delivery of network services with high data rate, processing and storage demands for a geographically widely dispersed user base.

B. Van Houdt and J.F. Pérez are with PATS Research Group, Department of Mathematics and Computer Science, University of Antwerp – IBBT, Antwerp, Belgium. Email: {benny.vanhoudt, juan-fernando.perez}@ua.ac.be

C. Develder, M. Pickavet and B. Dhoedt are with the Department of Information Technology (INTEC), Ghent University – IBBT, Ghent, Belgium. Email: {chris.develder, mario.pickavet, bart.dhoedt}@intec.ugent.be

Manuscript received August 6, 2008.
Revised manuscript received January 11, 2010.
C. Develder is supported by the Research Foundation – Flanders (FWO–Vl.) as a postdoctoral fellow.

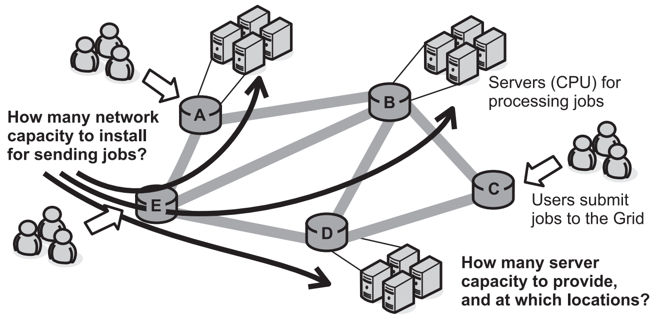

How many network capacity to install for sending jobs?
A
B
C
D
E
Servers (CPU) for processing jobs
Users submit jobs to the Grid
How many server capacity to provide, and at which locations?

Fig. 1. The Grid dimensioning problem involves both network and Grid resource dimensioning to cater for a given load of jobs submitted by users.

To maximize the fulfillment of that promise, some fundamental questions should be addressed, such as (re)designing the architecture of a flexible optical layer (e.g. evolving to Grid-OBS [2], [3], or some hybrid circuit/burst networks) and the development of appropriate routing and scheduling algorithms for these networks. Major differences with traditional network design originate from the anycast routing principle: Grid users generally do not care where exactly their jobs end up being executed, as long as they get executed timely. Hence, jobs can be sent off to 'any' suitable location and traffic volume is dependent on dimensions and locations of computation/storage resources, as well as the job scheduling algorithm. These Grid specific aspects give rise to multiple challenging research questions [4], [5], e.g. jointly optimizing not just (computational) Grid resources, but also the underlying optical network interconnecting them.

In this paper, we focus on the problem of Grid dimensioning, as sketched in Fig. 1. The input is a given network topology—the locations of the sites where jobs originate (or aggregation points where they are collected, e.g. points-of-presence of Grid service providers) and a (backbone) network interconnecting them—and the amount of Grid jobs generated at each of the sites. We want to find where to provide how much server capacity (esp. for computation), and the network dimensions required to process the submitted jobs. The aforementioned anycast principle complicates answering these questions with traditional algorithms, since we lack the complete so-called traffic matrix stating the amount of bandwidth exchanged between each (source, destination)-pair: the destination of a Grid job can be freely chosen by some job scheduling algorithm. For example, in Fig. 1 it is a priori not known which fraction of the jobs originating at site E will be sent to each of the server sites A, B and D.

In the subsequent Section II, we will describe a proposal to solve this Grid dimensioning problem and related work. One particular step of the dimensioning methodology requires calculating inter-site bandwidth, for which we propose an analytical mean field solution discussed in detail in Section III. We discuss a case study on a realistic European network scenario in Section IV and conclude in Section V.

II. Dimensioning optical Grids

A classical network design problem is dimensioning: how much capacity is needed for the network to be able to transport a given amount of traffic? Typically, this traffic is specified in a traffic matrix giving the amount of traffic $D_{i,j}$ flowing from site i to j (e.g. in Mbit/s). Well-known research literature on network dimensioning assumes this given traffic matrix D, yet in Grids this is not known a priori (cf. anycast principle). Moreover, for Grids, in addition to the network, also computational and storage capacity of the servers at the various Grid sites needs to be dimensioned. In this work, we focus on computational Grids: we need to determine the amount of processors (CPUs) to place at each of the chosen Grid sites.

A. Related work

For dimensioning (optical) networks without considering the Grid resources, a broad range of algorithms are available. The algorithms vary depending on the network technologies and topologies (e.g. single or multi-layer [6], with or without grooming [7]; for ring [8] or mesh networks), design criteria (e.g. survivability [9], availability), single or multi-period planning [10], single domain or hierarchical networks [11], etc. Yet, for dimensioning grids, the anycast routing principle gives rise to the problem of accurately estimating the (source,destination)-based traffic matrix these approaches all rely on.

In Grids also the computational and/or storage resources need to be dimensioned: how many servers need to be installed, and at which sites? The latter will have an impact on where jobs will end up being executed, i.e. the eventual traffic matrix, hence the network dimensions. It is clear that jointly determining both server and network dimensions is a very hard problem (even single-period network dimensioning for a given traffic matrix may already be NP-hard [12]). Therefore, we will propose a phased approach, dimensioning first the servers and then the network (see Section II-B).

Related work on dimensioning Grids is scarce. In [13] analytical ILP (integer linear programming) and heuristic approximations are used to cater for excess load: it is assumed that each of the Grid sites (dimensioned for the locally generated jobs) may suffer from overload, and network dimensions (number of wavelengths and fibers used) are determined by finding a global optimum over all single-site overload problems.

One way to deal with the unknown destination for Grid jobs is to assume that the fraction of jobs (originating at a particular site) going to a given computational Grid site is known, thus fixing a priori the arrival rates of jobs at each job execution site. This approach is taken in [14] (assuming OBS), where an analytical methodology known as reduced load fixed-point approximation [15] is used to dimension both network and computational resources.

In this paper however, we focus on a ‘clean slate’ or greenfield Grid dimensioning problem finding the complete Grid capacity required to meet a given Grid job arrival pattern. Also, we assume fully flexible scheduling strategies without any knowledge of a priori given probabilities for selecting a given destination site. Yet, since scheduling algorithms as such are not in the scope of this paper, we will assume fairly straightforward scheduling strategies, based on a single all-knowing scheduler, finding a free server for every arriving job based solely on the job's arrival time and duration, and server processing speed and occupation. For more advanced scheduling algorithms, including e.g. advance reservation concepts and QoS support, we refer to [16], [17].

B. An iterative dimensioning approach

To deal with the complex problem of calculating the required amount of Grid and network resources, we proposed an iterative approach to Grid dimensioning [**?**], comprising successive steps eventually leading to a traffic matrix, allowing traditional algorithms to solve the network dimensioning problem for the network technology of choice. This iterative approach can be summarized as follows:

D1. Out of the N Grid sites, find the N_s best server locations to install servers.
D2. Determine the amount of server capacity (number of CPUs) to install at each of the N_s chosen server locations.
D3. Calculate the amount of jobs $D_{i,j}$ sent from each originating site i to each of the destination sites j (being one of the N_s chosen locations).
D4. Calculate the network dimensions for the traffic demand matrix D from step D3.

In this approach, steps D1–D2 were solved analytically by respectively a fairly simple ILP solution and heuristic calculations, as described in detail in [?]. The ILP for step D1 aims to minimize the bandwidth used for transferring the Grid jobs to the servers, making some simplifying assumptions. In step D2, we use the well-known ErlangB formula to calculate the total amount of servers required, and use heuristics to distribute this server capacity over the locations chosen in step D1. For step D3, in previous work we resorted to (time consuming) simulations. For the final step D4, traditional algorithms for (optical) network dimensioning can be used, taking the traffic matrix D calculated in step D3 as input. It is important to stress that the network topology (and link capacities) will not influence step D3, meaning no routing issues are considered during this step. The link capacities are only determined in step D4 by making use of the traffic matrix D obtained in step D3 and the network topology.

In this paper, we introduce an analytical framework for step D3. When making some a priori assumptions about the distribution of jobs over the various Grid sites, fixed point approximation (FPA) has been successfully used for analytical dimensioning of the underlying optical network (e.g. [14] for the OBS case). Yet, if we want to model scheduling algorithms choosing any of the Grid sites having a free server at job arrival, the discrepancy between an FPA model's results and simulations is large [?]. In Section III, we introduce another analytical approach called mean field calculation. We will show that this method is practical for the Grid dimensioning case at hand and closely matches with (time consuming) simulations. Thus, the mean field solution can be used for step D3 and speed up the dimensioning cycle by avoiding simulations.

As stated before, the amount of jobs sent to a particular site also depends on the scheduling algorithm. In this paper, we consider two alternatives (*random* and *mostfree*, see further) to choose a server when a job arrives. In either case, if the job arrives at site i and this has a free server CPU at that time, this local CPU at site i will be chosen. Note that we consider that a job will occupy a single server CPU for the entire job duration. (We do not model job interdependencies, e.g. for user tasks comprising multiple jobs.)

III. A MEAN FIELD SOLUTION FOR INTER-SITE BANDWIDTH CALCULATION

The model as described in detail below, is a discrete-time model, where time is subdivided in so-called epochs of a fixed duration. A Grid site (recall Fig. 1) will be characterized by the number of servers it has (zero or more), and the amount of jobs arriving at this site. Each of the servers is assumed to be identical, and is able to process one job at a time. The analytical methodology used only works efficiently if the amount of different site characteristics is limited: sites will be partitioned into classes, where all sites of a particular class have the same number of servers and identical job arrival processes. We will show that, despite this at first sight severe limitation, the method is applicable in realistic scenarios.

We now describe the mean field (MF) solution, starting off with the assumed Grid network and job models. Subsequently, in Section III-B we outline the analytical framework for solving the case where all Grid sites are identical in job arrival process and server capacities. The general case, where Grid sites—as in most practical cases—differ in amount of traffic arriving and/or server capacities, is treated in the Appendix. In III-C we explain how to compute the demand matrix D from the mean field model results.

A. Grid model

We consider a grid network consisting of N sites, partitioned into K classes, assuming all sites belonging to the same class k have the same characteristics:

1) a class-k site has $C^{(k)}$ identical servers,
2) the inter-arrival times (IATs) of jobs originating at a class-k site are independent and identically distributed (i.i.d) and follow a discrete-time phase-type (PH) distribution with parameters $(\vec{\alpha}^{(k)}, T^{(k)})$ (cf. infra),
3) processing a job at a class-k site takes a geometric amount of time with mean $1/p^{(k)}$.

This class partitioning may seem to limit the applicability of the mean field solution discussed below. However—as we will illustrate in Section IV for realistic scenarios—clustering techniques can be used to achieve such partitioning into a limited number of classes.

The model is a discrete-time model where at each time epoch, three sequences of events occur:

S1. *Service completions:* each class-k occupied server becomes idle with probability $p^{(k)}$.

S2. *Arrivals:* at each site either 0 or 1 job arrives with a probability depending on the underlying phase of the arrival process at that particular site (see below; the model can easily be extended to batch arrivals of > 1 jobs). If a job arrives at a site with at least one local server (i.e., at the same site) available after step S1, the job is processed by a local server. Otherwise, the job becomes part of the pool of excess jobs.

S3. *Excess redistribution:* All excess jobs after step S2 are distributed among the servers that remained idle in step S2.

To redistribute j excess jobs over s idle servers in step S3, we consider two redistribution schemes: *mostfree* and *random*. Clearly, if $j > s$, all servers become occupied and we drop $j - s$ jobs. For $j \leq s$, the *mostfree* strategy will assign the j jobs one by one, each time selecting the site with the highest number of free servers (at the time of assignment). The *random* strategy simply selects the j servers at random among the s available ones, without considering the occupancy level of the site to which a server belongs.

With respect to the dropped jobs, we should add that as the number of Grid sites increases the job drop probability will decrease to zero. In actual fact the mean field model describes the limiting case as the number of sites goes to infinity and therefore its drop probability equals zero. For a finite, large number of sites job losses will be rare and could be even further reduced by adding finite buffers to each of the sites. In this case we might either decide (1) to forward a job only to another site when all the local servers are busy and the local buffer is full or (2) to buffer the job locally if none of the sites has an available server. The mean field model in this paper could be extended to capture both cases, though in the latter case the results would coincide as in the limiting case we have no losses.

The mean field analysis presented below computes exact results for the limiting system behavior (i.e. steady state) when the number of sites per class goes to infinity. However, the number of sites per class does not have to be identical: if the number of class-k sites is defined as $\gamma_k N$ (where $\sum_k \gamma_k = 1$), then the limiting behavior corresponds to letting N approach infinity. Our case study will show that for practical site counts (some tens to a few hundreds), the limit behavior matches quite well with simulations for a finite number of sites.

Before we proceed, let us briefly discuss the discrete-time PH arrival process. It can, among others, capture any IAT distribution with a finite support (i.e. with a finite number of possible outcomes) and many moment matching procedures have been developed such that real-life higher moment measurements can be easily incorporated in the process [18]. Let X_n be the IAT between the n^{th} and $n+1^{\text{st}}$ arrival. Formally, the PH process is a discrete-time renewal process (the IATs $(X_n)_{n\geq 0}$ are i.i.d.) characterized by a stochastic $1 \times h$ vector $\vec{\alpha}$ (i.e., a vector with non-negative entries that sum to one) and a sub-stochastic $h \times h$ matrix T (i.e., a matrix with non-negative entries whose row sums are smaller than or equal to one), with $h \geq 1$, such that for $s \geq 1$, $P[X_n = s] = \vec{\alpha} T^{s-1} \vec{\theta}$, where $\vec{\theta} = \vec{e}_h - T\vec{e}_h$, with $\vec{e}_h$ an $h \times 1$ vector with all its entries equal to one. For instance, setting $h = 1$ results in a Bernoulli arrival process (the discrete time variant of the Poisson process). Also, any mixture of h geometric distributions can be realized by a diagonal T matrix containing the geometric parameters and a stochastic vector $\vec{\alpha}$ holding the weights of each of the distributions.

In order to obtain a suitable h, vector $\vec{\alpha}$ and matrix T in practice, one typically starts by measuring the first few moments of the IATs and matches these using a PH-distribution as in [18]. In Section IV we will rely on the first three moments of the IAT distribution, which typically results in an order 2 phase-type distribution, i.e., $h = 2$. Thus, as demonstrated in Section IV-D, phase-type IATs allow us to change the variation of the IATs, while keeping the mean fixed. In case of a Poisson (or Bernoulli) process the variation is uniquely determined by its mean providing less flexibility.

The PH renewal process is very suitable for Markovian modeling environments as $\vec{\alpha}_s$ may be regarded as the probability that the IAT starts in phase s. Further, given that the phase at the current time instant equals s, the arrival process will be in phase s' at the next time instant with probability $[T]_{s,s'}$ without having an arrival, whereas $[\vec{\theta}\vec{\alpha}]_{s,s'}$ gives the probability that there is an arrival and the initial phase of the next IAT is s' (note: $[X]_{s,s'}$ is entry (s, s') of the matrix X). In our model each class-k site is fed by its own instance of a PH renewal process with parameters $(\vec{\alpha}^{(k)}, T^{(k)})$.

B. A mean field solution for the single class Grid network

We first consider a Markovian model for the single class Grid network ($K = 1$); as such we can temporarily drop the superscript $^{(k)}$. For example, if the Grid dimensioning approach (in step D2, Section II-B) equally distributes server capacity over the chosen server locations, all these locations are identical in terms of server/processing capacity, each site having C servers. If we also assume all server locations have the same job arrival process, this amounts to a single class grid network, using the aforementioned terminology.

The idea of the Markovian model is to associate $h \cdot (C + 2)$ states with each site. State $\langle i, j\rangle$, with $0 \leq i \leq C + 1$ and $1 \leq j \leq h$, indicates that i jobs are present at the site after step S2, while the arrival process is in phase j. Recall that after step S2 a site can hold $C + 1$ jobs if all its servers were busy before this step and a new (excess) job arrives. Given that we have N sites, we get a total of $h^N \cdot (C + 2)^N$ states, which clearly can become huge. However, the mean field computation will be restricted to matrices of size $h \cdot (C + 2)$ and therefore turns out to be very effective.

1) Step S1, service completions: Given that i of the C servers are busy, i' of them will become available with probability $s_{i,i-i'} = \binom{i}{i'} p^{i'} (1-p)^{i-i'}$. For further use, we define the $h(C+1) \times h(C+1)$ triangular matrix S as

$$S = \begin{bmatrix} s_{0,0} & 0 & \cdots & 0 \\ s_{1,0} & s_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ s_{C,0} & \cdots & s_{C,C-1} & s_{C,C} \end{bmatrix} \otimes I_h, \qquad (1)$$

where I_h is the identity matrix of size h (reflecting the fact that the phase of the arrival process is not influenced by the service completions) and $\otimes$ denotes the Kronecker product between matrices.

2) Step S2, job arrivals: Given that the PH renewal process is in state j, it will generate an arrival and go to state j' with probability $[\vec{\theta}\vec{\alpha}]_{j,j'}$, while with probability $[T]_{j,j'}$ a similar transition occurs without involving an arrival. We also define the $h(C+1)\times h(C+2)$ matrix A as

$$A = \begin{bmatrix} T & \theta\alpha & 0 & \dots & 0 \\ 0 & T & \theta\alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & T & \theta\alpha \end{bmatrix}. \tag{2}$$

3) Step S3, excess redistribution: The state transition at time t due to the redistribution of the excess jobs at a site is influenced by the vector $\vec{M}^N(t) = 1/N \cdot (a_0(t), a_1(t), \dots, a_C(t), a_{C+1}(t))$, where

- $a_i(t)$ for $i = 0 \dots C$ is the number of sites with i busy servers after step S2 that do not have an excess job, while
- $a_{C+1}(t)$ indicates the number of sites with an excess job (i.e., the total number of excess jobs at time t). All C servers of these sites are clearly occupied.

Thus, the i-th entry $\vec{M}^N_i(t)$ of the vector $\vec{M}^N(t)$ equals the fraction of sites holding i jobs (incl. excess job) after step S2.

First consider the *mostfree* strategy. Let $q_{i,i'}(\vec{M}^N(t))$ be the probability that a site receives $i'-i \geq 0$ excess jobs, given that it held $i \leq C$ jobs after step S2. For *mostfree*,

- the first $a_0(t)$ excess jobs will be assigned to the sites with all their servers available,
- the next $a_0(t)+a_1(t)$ excess jobs are forwarded to the sites that had either 0 or 1 busy server (after this step all the sites with 0 busy servers received two excess jobs, while those with 1 busy server received 1 excess job),
- this continues until all $a_{C+1}(t)$ jobs have been distributed among the free servers or until all servers are busy.

For ease of notation define $b_i(t) = \sum_{k=0}^{i} a_k(t)$ as the number of sites with at most i busy servers after step S2. Provided that we have enough free servers to support the excess jobs, we can find a c, with $0 \leq c < C$, such that

$$\sum_{k=0}^{c-1} b_k(t) < a_{C+1}(t) \leq \sum_{k=0}^{c} b_k(t), \tag{3}$$

which we denote as $c(\vec{M}^N(t))$ (for $a_{C+1}(t) = 0$, we set $c = 0$). In other words, all sites with $i \leq c(\vec{M}^N(t))$ busy servers after step S2 ($b_{c(\vec{M}^N(t))}(t)$ in total) will end up with at least $c(\vec{M}^N(t))$ jobs and some of them with $c(\vec{M}^N(t))+1$ jobs, after step S3. The fraction of these sites with $c(\vec{M}^N(t))$ jobs equals

$$\beta_{c(\vec{M}^N(t))} = \frac{\sum_{k=0}^{c(\vec{M}^N(t))} b_k(t) - a_{C+1}(t)}{b_{c(\vec{M}^N(t))}(t)}. \tag{4}$$

Thus, $i'-i \geq 0$ jobs are received by a site with i busy servers with probability $\beta_{c(\vec{M}^N(t))}$ whenever $i' = c(\vec{M}^N(t))$ and with probability $1-\beta_{c(\vec{M}^N(t))}$ for $i' = c(\vec{M}^N(t))+1$. If the number of free servers $\sum_{k=0}^{C-1} b_k(t)$ is insufficient to support the $a_{C+1}(t)$ jobs, we let $c(\vec{M}^N(t))$ equal C. In this case all the servers become occupied. This yields,

$$q_{i,i'}(\vec{M}^N(t)) = \begin{cases} 1-\beta_{c(\vec{M}^N(t))} & i < i' = c(\vec{M}^N(t))+1 \leq C, \\ \beta_{c(\vec{M}^N(t))} & i \leq i' = c(\vec{M}^N(t)) < C, \\ 1 & i = i' > c(\vec{M}^N(t)) \\ 1 & i' = C = c(\vec{M}^N(t)), \end{cases} \tag{5}$$

for $0 \leq i, i' \leq C$, where the third case indicates that no jobs are received when $i > c(\vec{M}^N(t))$. For further use, we also define the $h(C+2)\times h(C+1)$ matrix $Q(\vec{M}^N(t))$ as

$$Q(\vec{M}^N(t)) = \begin{bmatrix} q_{0,0}(\vec{M}^N(t)) & q_{0,1}(\vec{M}^N(t)) & \dots & q_{0,C}(\vec{M}^N(t)) \\ 0 & q_{1,1}(\vec{M}^N(t)) & \dots & q_{1,C}(\vec{M}^N(t)) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & q_{C,C}(\vec{M}^N(t)) \\ 0 & \dots & 0 & 1 \end{bmatrix} \otimes I_h, \tag{6}$$

where the 1 in the lower right corner indicates that sites with $C+1$ jobs after step S2 will end up with C jobs after step S3 (either due to a redistributed or a dropped job).

Next, consider the *random* redistribution strategy for the excess jobs, defining $\bar{Q}(.)$ analogously to *mostfree*'s $Q(.)$. Assume site s has i occupied servers. In total there are $f(\vec{M}^N(t)) = \sum_{k=1}^{C} a_{C-k}(t)k$ servers to choose from and $C-i$ of them belong to site s. Therefore the probability that $0 \leq i' \leq C-i$ excess jobs are assigned to site s, equals

$$\bar{q}_{i,i+i'}(\vec{M}^N(t)) = \frac{\binom{C-i}{i'}\binom{f(\vec{M}^N(t))-(C-i)}{a_{C+1}(t)-i'}}{\binom{f(\vec{M}^N(t))}{a_{C+1}(t)}}, \tag{7}$$

provided that $f(\vec{M}^N(t)) \geq a_{C+1}(t)$, otherwise we have for all i that $\bar{q}_{i,C}(\vec{M}^N(t)) = 1$.

4) Combining steps S3, S1 and S2: To obtain a useful discrete time Markov chain description of the system, we will observe it at each time epoch immediately after step S2 and before step S3. Given the state $\langle i,j \rangle$ of site s at time t (with i the number of jobs, and j the PH phase, see above), we can obtain its system state at time $t+1$, which depends on $\vec{M}^N(t)$, via the transition matrix $\mathcal{K}(\vec{M}^N(t))$ defined as

$$\mathcal{K}(\vec{M}^N(t)) = Q(\vec{M}^N(t)) \cdot S \cdot A, \tag{8}$$

for the *mostfree* strategy. (For the *random* strategy, we simply replace $Q(.)$ by $\bar{Q}(.)$ to obtain $\bar{\mathcal{K}}(\vec{M}^N(t))$.) Since the state evolution of different sites are correlated, the transition matrix of the entire system is hard to express. Luckily, the mean field computation only requires $\mathcal{K}(\vec{M}^N(t))$ for N going to infinity and will allow us to express $\vec{M}^N(t+1)$ as a simple function of $\vec{M}^N(t)$ using $\mathcal{K}(\vec{M}^N(t))$ for N large.

5) Fast computation of $\vec{M}^N(t)$ for large N: The main objective of the grid network model is to be able to calculate inter-site rates (cf. traffic matrix). Given the rate r_{kj} of excess jobs of a particular class-k site processed by any of the $\gamma_j N$ class-j sites, the traffic rate from a single class-k site to a single class-j site equals $r_{kj}/(\gamma_j N)$. For the single class model, r_{11} is given by the last component of the occupancy measure $\vec{M}^N(t)$ for t large enough: $\vec{M}^N_{C+1}(t)$ represents the proportion of the sites that hold an excess job just prior to the redistribution step.

To find $\vec{M}^N(t)$, we apply a generic framework of interacting objects introduced in [19]. Under some fairly mild conditions, as the number of objects becomes large, the occupancy measure of the system converges to a deterministic dynamical system,—the mean field—that has the same dimension as a single object. In our Grid model, we associate a single object with each site, such that the matrix $\mathcal{K}(\vec{M}^N(t))$ (or $\bar{\mathcal{K}}(.)$, the discussion below applies to both) describes the evolution of a single object as a function of the occupancy measure $\vec{M}^N(t)$.

The convergence result [19, Theorem 4.1] is proven to hold if there exists a $\mathcal{K}(\vec{M}(t))$ matrix, such that for each entry $[\mathcal{K}(\vec{M}^N(t))]_{s,s'}$, the set of functions $\{[\mathcal{K}(\vec{M}^N(t))]_{s,s'}, N \geq 1\}$ converges uniformly to $[\mathcal{K}(\vec{M}(t))]_{s,s'}$ on the set of all possible occupancy vectors $\vec{M}(t)$. For the *mostfree* model this convergence is immediate as the $\mathcal{K}(\vec{M}^N(t))$ matrices are independent of N (to see this, simply divide all the $a_k(t)$ and $b_k(t)$ appearing in $Q(\vec{M}^N(t))$ by N). For the *random* strategy, we define $\bar{q}_{i,i+i'}(\vec{M}(t))$ as

$$\bar{q}_{i,i+i'}(\vec{M}(t)) = \binom{C-i}{i'} \left(\frac{\vec{M}_{C+1}(t)}{f(\vec{M}(t))}\right)^{i'} \left(1 - \frac{\vec{M}_{C+1}(t)}{f(\vec{M}(t))}\right)^{C-i-i'}, \quad (9)$$

with $f(\vec{M}(t)) = \sum_{j=1}^{C} \vec{M}_{C-j}(t) j$, if $\vec{M}_{C+1}(t) \leq f(\vec{M}(t))$, otherwise $\bar{q}_{i,C}(\vec{M}(t)) = 1$ for all i. Finally, define $\bar{Q}(\vec{M}(t))$ analogously to (6). It is not hard to show that the set of functions $\{[\bar{\mathcal{K}}(\vec{M}^N(t))]_{s,s'}, N \geq 1\}$, for any s, s' converges uniformly to $[\bar{\mathcal{K}}(\vec{M}(t))]_{s,s'} = [\bar{Q}(\vec{M}(t))SA]_{s,s'}$ on the set of all occupancy vectors $\vec{M}(t)$.

Next, [19] requires $\mathcal{K}(\vec{M}(t))$ to be continuous in $\vec{M}(t)$, which is clearly the case for both *mostfree* and *random*. Thus, due to [19, Theorem 4.1], the following convergence result for the mean field applies. Define the $1 \times h(C+2)$ vector $\vec{\mu}(0)$ as $(\vec{\alpha}, 0, \ldots, 0)$, where $\vec{\alpha}$ is the initial vector of the PH renewal process (i.e. system empty at 0) and let

$$\vec{\mu}(t+1) = \vec{\mu}(t)\mathcal{K}\left(\vec{\mu}(t)\left(I_{C+2} \otimes \vec{e}_h\right)\right). \quad (10)$$

Then, for any t, *almost surely*,

$$\lim_{N\to\infty} \vec{M}^N(t) = \vec{\mu}(t)\left(I_{C+2} \otimes \vec{e}_h\right). \quad (11)$$

Thus, to compute the mean field at time t, it suffices to perform t matrix multiplications with matrices of size $h(C+2)$ only.

As $h = 2$ often suffices to match up to three moments of the IAT distribution, $h(C+2)$ will be fairly small, resulting in a fast computation of $\vec{\mu}(t)$. Since our interest lies mainly in steady state behavior (i.e. t large), we will iteratively compute $\vec{\mu}(t)$ until $||\vec{\mu}(t) - \vec{\mu}(t-1)|| < \epsilon$, for small ϵ. The computation time can be reduced by selecting a different initial vector $\vec{\mu}(0)$, that is closer to $\vec{\mu}(t)$ for t large; e.g., investigating excess traffic rates for various system loads, we could use the steady state of the previous load as an initial vector for the next case.

C. Calculating the demand matrix D

In the previous section, the mean field approach for the single class case was explained, allowing to calculate the site occupancy measure $\vec{M}^N(t)$. Similarly, in the Appendix we detail how to calculate the occupancy measure $\vec{M}^{N,(k)}(t)$ for the case of multiple site classes. There, $\vec{M}^{N,(k)}_i(t)$ represents the proportion of class-k sites holding i jobs after step S2 ($0 \leq i \leq C^{(k)}+1$). Thus, the proportion of class-k sites with excess jobs equals $M^{(k)}_{C^{(k)}+1}(t)$, for t large. As all class-k sites are identical, $\vec{M}^{N,(k)}_i(t)$ is also the percentage of time in which a class-k site has an excess job. Therefore it equals the excess rate of a class-k site. With $\lambda^{(k)}$ the mean job arrival rate at a class-k site, the rate of excess jobs processed by a class-k site, denoted as $\lambda^{(k)}_{exc}$, is found as the rate at which a class-k site completes jobs minus the rate of completed jobs that originated in this site; hence,

$$\lambda^{(k)}_{exc} = \vec{\mu}^{(k)}(t) Q^{(k)}(\vec{M}^{(k)}(t)) \begin{bmatrix} 0 \\ p^{(k)} \\ 2p^{(k)} \\ \vdots \\ C^{(k)}p^{(k)} \end{bmatrix} \otimes \vec{e}_{h^{(k)}} - (\lambda^{(k)} - \vec{M}^{(k)}_{C^{(k)}+1}(t)), \quad (12)$$

for t large.

As the probability that an excess job receives service in a class-j site is independent of its type under the *mostfree* and *random* strategy, the rate $r_{k,j}$ of excess jobs of a class-k site served by any class-j site can be computed as

$$r_{k,j} = \vec{M}^{(k)}_{C^{(k)}+1}(t) \frac{\lambda^{(j)}_{exc}}{\sum_{s=1}^{K} \lambda^{(s)}_{exc}} = \lambda^{(j)}_{exc} \frac{\vec{M}^{(k)}_{C^{(k)}+1}(t)}{\sum_{s=1}^{K} M^{(s)}_{C^{(s)}+1}(t)},$$

for t large. From these inter-class rates, the demand matrix D can be easily calculated: the rate from a site s of class k to a site d of class j is $D_{s,d} = r_{k,j}/(\gamma_j N)$ (with $\gamma_j N$ the number of class-j sites).

IV. Numerical results

In this section we first consider two simple Grids in order to analyze the effect of the scheduling algorithm on the performance of the Grid, in terms of the spill rates. Next, we test the mean field model by considering a realistic European network scenario.

A. The effect of the scheduling algorithm

We first consider a Grid consisting of many sites partitioned in two classes. All the sites have 20 servers and the same arrival process, a Bernoulli process with mean IAT equal to 30 seconds. Class-2 sites represent only 1% of the total number of sites and their load, given by $\rho^{(2)} = \frac{\lambda^{(2)}}{\mu^{(2)} \cdot C^{(2)}}$, is equal to 0.95, i.e. they are heavily loaded. The remaining 99% of the sites are of class 1 and a load between 0.1 and 0.95 will be considered. When their load is equal to 0.95, all the sites in the Grid are identical. Fig. 2 shows the total spill rate at class-2 sites, and the rate at which these spilled jobs are sent and processed at class-1 and class-2 sites. We observe that when the load of the class-1 sites is low, the *mostfree* algorithm allocates almost every excess job from a class-2 site to a class-1 site. This is the case for loads up to 0.7 in this scenario. On the other hand, the *random* policy assigns a significant fraction of excess jobs to the heavily-loaded class-2 sites. Although this has little influence in the total spill rate of the class-2 sites for low and mid loads, for loads above 0.75 the *mostfree* policy offers a reduction in the spill rate. In fact, the total spill rate under this policy can be up to 20% smaller than under the *random* scheduling. As expected, when both class-1 and class-2 sites have the same load, i.e. $\rho^{(1)} = 0.95$, the spill rate from class-2 sites toward both sites of both classes are equal, while the *mostfree* policy still causes a significantly smaller spill rate than the *random* allocation.

Next, we consider a single-class Grid and compute the spill rate for different values of C, the number of servers per site. The results are included in Fig. 3, where the difference between these two policies becomes apparent at high loads. We also find that the maximum reduction in spill probability caused by using the *mostfree* policy is around 15% for $C = 5$, near to 20% for $C = 20$ and above 22% for $C = 100$. Therefore we see an increment in the maximum relative difference in spill rate as the number of servers per site increases. However, from Fig. 3, we also observe that the load range for which the *mostfree* policy outperforms the *random* allocation decreases with the number of servers.

B. European Grid use case

The preconditions to allow our mean field methodology are: (i) the job inter-arrival time (IAT) distribution should be modeled as a discrete-time phase-type (PH) distribution, (ii) the grid sites should be partitioned in a limited number of classes, and (iii) the number of Grid sites should be large enough. Conditions (ii)–(iii) are required because the mean field assumes an infinite number of sites per class. Hence, mean field results are expected to be closer to those of the finite system when sites are partitioned into a few classes each with a significant number of sites. As discussed in III-A, condition (i) is not really limiting, since many real-world traces can be matched with a limited number of phases (keeping the analytical model compact) using moment-matching procedures.

With respect to (iii), realistic use cases for Grid dimensioning would comprise from some tens to a couple of hundreds of sites. These numbers are still acceptable for the methodology to be practical as will be clear from the subsequent case studies for $N = 100$ sites and five classes. With respect to the computation times, we found that the arrival process variability affects the number of iterations required for convergence of $\vec{\mu}(t)$, while the overall load seems to have little effect. For this case study and with $\epsilon = 10^{-10}$, the computation times varied from one to ten minutes. These times can be further reduced, especially for the cases requiring more iterations, by initializing the system in the following manner: let $\pi_j^{(k)}$ be the stationary probability of having j busy servers in an

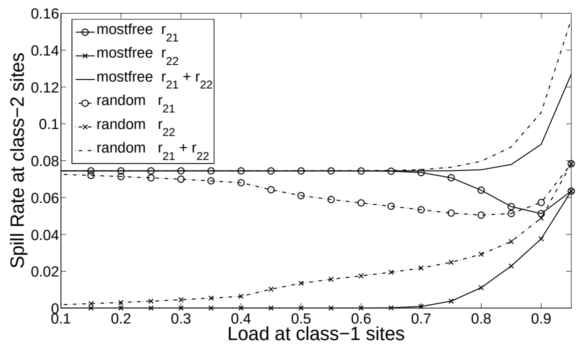


Fig. 2. Mean field results for a two-class Grid, with variable load for class-1 sites.

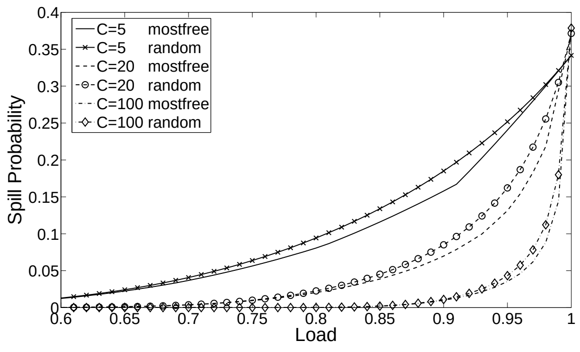


Fig. 3. Mean field results for a single-class Grid, with variable load and number of servers per site.

M/M/$C^{(k)}$/$C^{(k)}$ queue, and let $\tau^{(k)}$ be the stationary distribution of the PH arrival process at station k, i.e., $\tau^{(k)} = \tau^{(k)}\left(T^{(k)} + \theta^{(k)}\alpha^{(k)}\right)$ and $\tau^{(k)}\vec{e}_{h^{(k)}} = 1$. Then, by setting $\mu^{(k)}(0) = (\pi_0^{(k)}, \pi_1^{(k)}, \ldots, \pi_{C^{(k)}}^{(k)}) \otimes \tau^{(k)}$ we obtained a reduction of up to 80% in the number of iterations while the computation times decreased to less than four minutes. Note that simulation running times in our case study differ by an order of magnitude, amounting to several hours (e.g. for the case study of Fig. 4, simulating 10^7 time units took close to two hours).

The major limitation at first sight seems to be constraint (ii). However, looking at real world traces, many sites show similar behavior, which allows clustering the various sites into a limited number of classes. This can be achieved by the *K-means* clustering method [20], where each site is described by a set of N variables called descriptors: (C1) Select K points in the N-dimensional space as centroids c_k ($k = 1, \ldots, K$); (C2) Form clusters C_k by assigning each site s to the closest centroid c_k; (C3) Recalculate c_k as the N-dimensional mean over C_k; (C4) If any c_k changed in C3, go back to C2. We aim at characterizing each of the clusters with a PH distribution matching the first three moments of the IAT distribution. Hence, we chose as site descriptors: the first non-central moment, the squared coefficient of variation (SCV) and the third normalized moment (n_3) of the IAT distribution. Let m_i be the i^{th} non-central moment, then define SCV $= \frac{m_2}{m_1^2} - 1$ and $n_3 = \frac{m_3}{m_2 m_1}$. The reason to prefer SCV and n_3 rather than m_2 and m_3 is that they are not affected by the units in which the variables are measured. As the IAT distribution is based on real traces, we rely on the sample moments given by $\bar{m}_i = \frac{1}{S}\sum_{j=1}^{S} x_j^i$, where each x_j corresponds to one of the IAT samples, with S samples in total.

For our case studies, we used traces from a real-world EGEE/LCG Grid, deployed in Europe in the frame of the Large Hadron Collider (LHC) experiments at CERN in Geneva and the Enabling Grids for E-sciencE (EGEE) project [21]. We collected Grid-wide job arrival logs, recording the job arrival rate at 58 sites over a one month period. After screening, we left out 8 sites because of lack of data to allow reliable statistical analysis. We used the clustering approach above, and partitioned the sites into $K = 5$ classes. To characterize each site class, we used the average moments over the cluster's sites. For each class we used the method in [18] to match the first 3 moments of the job IATs with a PH model with $h = 2$ phases (except for class 2, whose very small SCV causes matching for $h = 2$ to be restricted to the first 2 moments [18]). Table I summarizes the class descriptors. It is important to note that these characteristics greatly vary, ranging from low to high arrival rates and from small to large variability. To challenge the mean field method, we considered a case study with $N = 100$ Grid sites, respecting the proportion of each server class as observed in the EGEE/LCG trace.

TABLE I
CHARACTERISTICS OF THE 5 SITE CLUSTERS

Class	Mean IAT (s)	SCV	n_3	% Sites	C
1	29.75	136.45	2207.08	10%	150
2	77.24	83.40	488.69	46%	100
3	3696.46	0.46	5.73	6%	5
4	458.08	10.35	60.83	28%	10
5	1870.45	2.95	10.05	10%	10

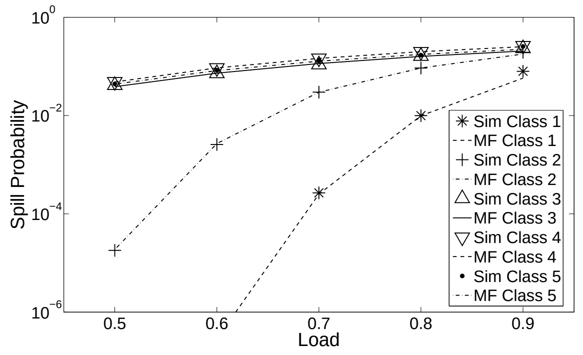
C. Varying the Grid resource load

First, consider varying the Grid system load ρ, i.e. setting each class-k site's load to $\rho^{(k)} = \rho$, with $\rho^{(k)} = \frac{\lambda^{(k)}}{\mu^{(k)} \cdot C^{(k)}}$ (where for a class-k site $\lambda^{(k)}$ is the average job arrival rate, $\mu^{(k)}$ the average job processing rate at a class-k site, and $C^{(k)}$ the number of servers). Given typical load values in network design, we studied $\rho \in [0.5, 0.9]$. We assumed $N = 100$ sites in total, comprising the $K = 5$ classes as outlined in Table I. The number of servers at each site is chosen to obtain the target ρ, setting the average service time $1/p^{(k)}$ for each site of class k as $1/p^{(k)} = \rho C^{(k)}\, \mathrm{E}[\mathrm{IAT}^{(k)}]$, with $\mathrm{E}[\mathrm{IAT}^{(k)}] = 1/\lambda^{(k)}$ the average job inter-arrival time (IAT) for class k.

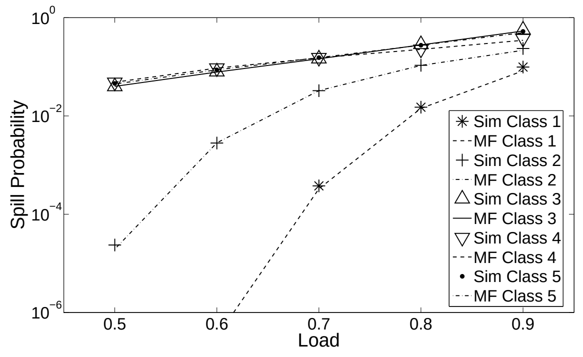
To evaluate the mean field methodology, we compared the results with the outcome of simulations. For this we implemented a discrete-event simulator and calculated the inter-site rates $D_{s,d}$ for each source site s and destination site d. To comprehensively present the results, the graphs will show for each class k the proportion of jobs sent to remote sites, i.e. the spill probability

$$P_{\text{spill},k} = \left(\sum_{s \in \text{class}k} \sum_{d \neq s} D_{s,d}\right) \Big/ \left(\sum_{s \in \text{class}k} \sum_{d} D_{s,d}\right).$$

We compared analytical results with simulations for both *random* and *mostfree* scheduling strategies. The graphs of Fig. 4 show that in both cases the analytical and simulation results match very well. For the whole load range, the analytically calculated spill rates fall well within the 95% confidence interval (not shown on the graphs for the sake of clarity) on simulations' spill rates (even though discrepancy increases for $\rho = 0.9$). Looking at the numerical values, we note that the discrepancy between analytical and simulation results is largest for classes 1 and 2 (but still less than the standard error on the simulation results; the standard error for a particular spill probability for class-k is given by $\text{stderr}^{(k)} = \sigma^{(k)}/(\gamma_k N)$ with $\sigma^{(k)}$ the variance on the spill probabilities for the $\gamma_k N$ class-k sites.) This can be explained by the large SCV on the job IAT in these site classes (see Table I). Note that—as expected—the *mostfree* strategy achieves lower spill probabilities than *random*, esp. for high loads ($\rho > 0.7$).



(a) *mostfree* scheduling



(b) *random* scheduling

Fig. 4. Simulation results match well with analytical mean field, for variable Grid resource load. Note that the curves for classes 3–5 overlap to great extent.

D. Varying variance on job IAT

Having established the close match between analytical mean field and simulation results over a broad load range, we investigated the impact of the variability of the job inter-arrival times. Hence, we fix load $\rho = 0.8$, but changed the SCV. For increasing variability on the job IATs, we expect higher $P_{\text{spill},k}$.

Figure 5 shows that even for larger SCV, the simulation results match the analytical results very well. As noted before, in terms of spill probability, *mostfree* outperforms *random* scheduling, but the amount seems dependent on job IAT variability. As expected, the overall spill probability (over all jobs, regardless of site class) increases with growing SCV.

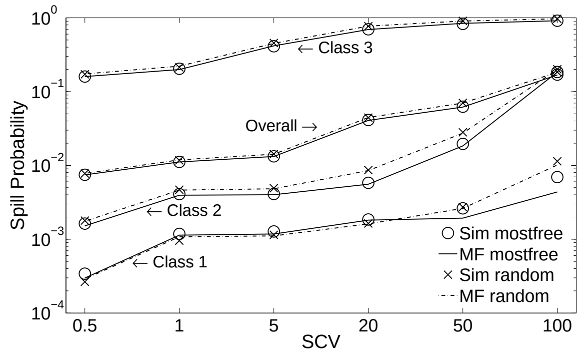


Fig. 5. Comparison of *mostfree* and *random* scheduling for different SCV of the inter-arrival distribution. Note that only classes 1–3 are shown, since results for classes 4 and 5 overlap with those of class 3.

V. Conclusion

Grid dimensioning involves answering the question how many servers to provide, where to place them, and which network to install for interconnection of server sites and users. Compared to traditional (optical) network dimensioning, Grids differ in two aspects. First, not only network but also server capacity needs to be dimensioned. Second, the (source,destination)-based traffic matrix—necessary for traditional dimensioning algorithms—is unknown, and depends on the Grid job scheduling algorithm.

This paper outlined a step-wise Grid dimensioning approach, resorting to traditional algorithms for network dimensioning. To calculate the required traffic matrix, we proposed an analytical mean field solution technique, avoiding time consuming simulation. The preconditions to allow our mean field methodology are: (i) job arrival should be modeled as a discrete-time phase-type (PH) distribution, (ii) grid sites should be partitioned in a limited number of classes (characterized by server capacity and job arrival parameters), (iii) the number of grid sites N should not be too small. Condition (i) is not really limiting, since PH models can match real world traces up to several higher moments already with a small number of phases ($h = 2$ for our case studies). Condition (ii) can be met by clustering real world sites into a limited number of classes ($K = 5$ based on a real world log comprising 50 sites). Studies showed that with respect to (iii), for a realistic number of sites in the range of some tens to a couple of hundreds, the analysis very well matches simulation results.

Feasibility of the mean field solution was illustrated by a case study for $N = 100$ sites. It showed a close correspondence between analysis and simulation for a broad range of loads (0.5 up to 0.9) and a varying degree of variance on the job inter-arrival times.

Appendix
Mean field solution for multi-class Grids

When not all sites have the same amount of servers, the model is a multiple class grid network. The single class grid network case can relatively easily be extended to the multi-class setting, because the Markov chain associated with each object in [19] is allowed to be reducible as explained below. The framework [19] applies to any system consisting of N objects, with N large, that are each characterized by a transition matrix $\mathcal{K}(\vec{M}^N(t))$. This remains true if the the state space of this transition matrix can be partitioned into K classes such that $\mathcal{K}(\vec{M}^N(t))$ can be written as a block diagonal matrix:

$$\mathcal{K}(\vec{M}^N(t)) = \begin{bmatrix} \mathcal{K}^{(1)}(\vec{M}^N(t)) & 0 & \dots & 0 \\ 0 & \mathcal{K}^{(2)}(\vec{M}^N(t)) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathcal{K}^{(K)}(\vec{M}^N(t)) \end{bmatrix}. \quad (13)$$

Because no transitions are possible between states belonging to different classes, the state of an object which belongs to class k at time $t = 0$ will always remain in that class. Hence, we let $\mathcal{K}^{(k)}(\vec{M}^N(t))$ characterize the transitions of a class-k site and define $\vec{\mu}(0)$, the system state at $t = 0$, such that $\gamma_k N$ of the N sites start in a class-k state.

Let $\vec{M}^{N,(k)}(t) = (a_0^{(k)}(t), \dots, a_C^{(k)}(t), a_{C+1}^{(k)}(t))/\gamma_k N$ be the occupancy measure of the class-k sites: $\vec{M}_i^{N,(k)}(t)$ represents the proportion of class-k sites holding i jobs after step S2 ($0 \leq i \leq C^{(k)} + 1$). The overall occupancy $\vec{M}^N(t)$ equals

$$\vec{M}^N(t) = \left(\gamma_1 \vec{M}^{N,(1)}(t), \dots, \gamma_K \vec{M}^{N,(K)}(t)\right).$$

A. Computing $Q^{(k)}(\vec{M}^N(t))$

To compute the mean field, we first need an expression for $\mathcal{K}^{(k)}(\vec{M}^N(t))$, the transition matrix of the class-k sites, given that the overall occupancy measure is $\vec{M}^N(t)$. As arrivals and service completions are not affected by the presence of multiple classes we still have $\mathcal{K}^{(k)}(\vec{M}^N(t)) = Q^{(k)}(\vec{M}^N(t)) S^{(k)} A^{(k)}$. The $Q^{(k)}(\vec{M}^N(t))$ matrices have the same form as in (6), except that the expressions for $q_{i,i'}(\vec{M}^N(t))$ and $\bar{q}_{i,i'}(\vec{M}^N(t))$ require some modifications as all the sites influence a class-k site and not just the other class-k sites.

1) Random scheduling: For the *random* strategy case one finds

$$\bar{q}_{i,i+i'}^{(k)}(\vec{M}^N(t)) = \frac{\binom{C^{(k)}-i}{i'} \binom{f(\vec{M}^N(t)) - (C^{(k)}-i)}{\sum_{k=1}^K a_{C^{(k)}+1}^{(k)}(t) - i'}}{\binom{f(\vec{M}^N(t))}{\sum_{k=1}^K a_{C^{(k)}+1}^{(k)}(t)}},$$

with $f(\vec{M}^N(t)) = \sum_{k=1}^K \sum_{s=1}^{C^{(k)}} s a_{C^{(k)}-s}^{(k)}(t)$.

2) Mostfree scheduling: Similar to Step S3 in Section III-B, we start by defining $b_i^{(k)}(t)$ as the number of class-k sites with at most i busy servers after step S2 at time t; then $b_{C^{(k)}-i}^{(k)}(t)$ denotes the number of class-k sites with at least i free servers. Let $a^T(t) = \sum_{k=1}^K a_{C^{(k)}+1}^{(k)}(t)$ denote the total number of excess jobs after step S2. Finally, without loss of generality, label the K classes such that $C^{(1)} \geq C^{(2)} \geq \dots \geq C^{(K)}$.

Provided that there are enough free servers at time t to support the excess jobs, we have $a^T(t) \leq \sum_{i=1}^{C^{(1)}} \sum_{k=1}^K b_{C^{(k)}-i}^{(k)}(t)$, where $b_i^{(k)} = 0$ for $i < 0$. Hence, for $a^T(t) > 0$, there exists a $0 < d \leq C^{(1)}$ such that

$$\sum_{i=d+1}^{C^{(1)}} \sum_{k=1}^K b_{C^{(k)}-i}^{(k)}(t) < a^T(t) \leq \sum_{i=d}^{C^{(1)}} \sum_{k=1}^K b_{C^{(k)}-i}^{(k)}(t),$$

which we denote as $d(\vec{M}^N(t))$ (for $a^T = 0$, we set $c = C^{(1)}$). For $K = 1$, $c(\vec{M}^N(t))$ as defined in (3) equals $C^{(1)} - d(\vec{M}^N(t))$. The value of $d(\vec{M}^N(t))$ corresponds to the highest number of free servers found in any site after step S3. Thus, any class-k site has at least $C^{(k)} - d(\vec{M}^N(t))$ busy servers after step S3. Hence, sites that had more than $d(\vec{M}^N(t))$ free servers after step S2, received one or more excess jobs such that exactly $d(\vec{M}^N(t))$ or $d(\vec{M}^N(t)) - 1$ free servers remain. Similar to (4), the fraction of sites with $d(\vec{M}^N(t))$ jobs is

$$\gamma_{d(\vec{M}^N(t))} = \frac{\sum_{i=d(\vec{M}^N(t))}^{C^{(1)}} \sum_{k=1}^K b_{C^{(k)}-i}^{(k)}(t) - a^T(t)}{\sum_{k=1}^K b_{C^{(k)}-d(\vec{M}^N(t))}^{(k)}(t)}.$$

Notice, for $K = 1$, we have $\gamma_{d(\vec{M}^N(t))} = \beta_{c(\vec{M}^N(t))}$. If the number of free servers $\sum_{i=1}^{C^{(1)}} \sum_{k=1}^K b_{C^{(k)}-i}^{(k)}(t)$ is insufficient to support the $a^T(t)$ jobs, we let $d(\vec{M}^N(t))$ equal d. In this case all the servers become occupied. This yields, for the class-k sites, for $k = 1, \dots, K$

$$q_{i,i'}^{(k)}(\vec{M}^N(t)) = \begin{cases} 1 - \gamma_{d(\vec{M}^N(t))} & i < i' = C^{(k)} - d(\vec{M}^N(t)) + 1 \leq C^{(k)}, \\ \gamma_{d(\vec{M}^N(t))} & i \leq i' = C^{(k)} - d(\vec{M}^N(t)) < C^{(k)}, \\ 1 & i = i' > C^{(k)} - d(\vec{M}^N(t)) \\ 1 & i' = C^{(k)} = C^{(k)} - d(\vec{M}^N(t)), \end{cases} \quad (14)$$

for $0 \leq i, i' \leq C^{(k)}$, where the third case indicates that no jobs are received when $i > C^{(k)} - d(\vec{M}^N(t))$.

B. Computing the mean field

For the *mostfree* case $\mathcal{K}^{(k)}(\vec{M}(t)) = \mathcal{K}^{(k)}(\vec{M}^N(t))$, for all N, whereas for the *random* setting, the uniform limit $\bar{\mathcal{K}}^{(k)}(\vec{M}(t))$ is obtained in exactly the same manner as in the single class model (i.e., the hypergeometric probabilities converge to binomial probabilities). Due to [19, Theorem 4.1], we may compute the mean field as follows:

$$\vec{\mu}^{(k)}(t+1) = \vec{\mu}^{(k)}(t) \mathcal{K}^{(k)} \Big(\gamma_1 \mu^{(1)}(t) \big(I_{C^{(1)}+2} \otimes \vec{e}_{h^{(1)}}\big), \dots, \gamma_K \mu^{(K)}(t) \big(I_{C^{(K)}+2} \otimes \vec{e}_{h^{(K)}}\big)\Big), \quad (15)$$

for all k, with $\mu^{(k)}(0) = (\alpha^{(k)}, 0, \dots, 0)$ (where $(\alpha^{(k)}, T^{(k)})$ characterizes the PH renewal process of a class-k site). The class-k occupancy measure equals

$$\lim_{N \to \infty} \vec{M}^{N,(k)}(t) = \mu^{(k)}(t) (I_{C^{(k)}+2} \otimes \vec{e}_{h^{(k)}}).$$

ACKNOWLEDGMENT

Part of the work described in this paper was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe), a Network of Excellence funded by the European Commission through the 7th ICT-Framework Programme, and the also EC-funded IST Phosphorus project.

REFERENCES

[1] G. Lawton, "Moving the OS to the web," *IEEE Computer*, vol. 41, no. 3, pp. 16–19, Mar. 2008.

[2] F. Farahmand *et al.*, "A multi-layered approach to optical burst-switched based grids," in *Proc. 2nd Int. Conf. on Broadband Networks (Broadnets 2005)*, Oct. 2005, pp. 1050–1057.

[3] M. De Leenheer *et al.*, "A view on enabling-consumer oriented grids through optical burst switching," *IEEE Commun. Mag.*, vol. 44, no. 3, pp. 124–131, Mar. 2006.

[4] D. Simeonidou *et al.*, "Dynamic optical-network architectures and technologies for existing and emerging grid services," *J. Lightwave Technology*, vol. 23, no. 10, pp. 3347–3357, Oct. 2005.

[5] M. De Leenheer, C. Develder, T. Stevens, B. Dhoedt, M. Pickavet, and P. Demeester, "Design and control of optical grid networks (invited)," in *Proc. 4th Int. Conf. on Broadband Networks (Broadnets 2007)*, Raleigh, NC, Sep. 2007.

[6] H. Höller and S. Voß, "A heuristic approach for combined equipment-planning and routing in multi-layer SDH/WDM networks," *European J. of Operational Research*, vol. 171, no. 3, pp. 787–796, Jun. 2006.

[7] K. Zhu, H. Zang, and B. Mukherjee, "A comprehensive study on next-generation optical grooming switches," *IEEE J. on Selected Areas in Communications*, vol. 21, no. 7, pp. 1173–1186, Sep. 2003.

[8] O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618–630, Oct. 2000.

[9] D. Colle *et al.*, "Data-centric optical networks and their survivability," *IEEE J. Selected Areas in Communications*, vol. 20, no. 1, pp. 6–20, Jan. 2002.

[10] M. Pickavet and P. Demeester, "Long-term planning of WDM networks: A comparison between single-period and multi-period techniques," *Photonic Network Communications*, vol. 1, no. 4, pp. 331–346, Dec. 1999.

[11] A. Bley, T. Koch, and R. Wessäly, "Large-scale hierarchical networks: How to compute an optimal architecture?" in *Proc. 11th Int. Telecommun. Network Strategy and Planning Symposium (Networks 2004)*, Vienna, Austria, 13-16 Jun. 2004.

[12] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 684–696, Oct. 1996.

[13] P. Thysebaert, F. De Turck, B. Dhoedt, and P. Demeester, "Using divisible load theory to dimension optical transport networks for grid excess load handling," in *Proc. Int. Conf. on Autonomic and Autonomous Systems & Int. Conf. on Networking and Systems (ICAS/ICNS 2005)*, Papeete, Tahiti, 23–28 Oct. 2005.

[14] M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, and P. Demeester, "Erlang reduced load model for optical burst switched grids," in *Proc. 3rd Int. Conf. on Networking and Services (ICNS 2007)*, Athens, Greece, 19–25 June 2007.

[15] Z. Rosberg, H. Vu, M. Zukerman, and J. White, "Blocking probabilities of optical burst switching networks based on reduced load fixed point approximations," in *Proc. 22nd Annual Joint Conf. of the IEEE Computer and Commun. Societies (INFOCOM 2003)*, San Francisco, CA, USA, 30 Mar. – 3 Apr. 2003, pp. 2008–2018.

[16] C. de Waal *et al.*, "D5.4 - support for advance reservations in scheduling and routing," IST Phosphorus project deliverable (online: http://www.ist-phosphorus.eu/deliverables.php), June 2007.

[17] E. Varvarigos *et al.*, "D5.2 - QoS-aware resource scheduling," IST Phosphorus project deliverable (online: http://www.ist-phosphorus.eu/deliverables.php), Sept. 2007.

[18] M. Telek and A. Heindl, "Matching moments for acyclic discrete and continuous phase-type distributions of second order," *International Journal of Simulation Systems, Science & Technology*, vol. 3, pp. 47–57, 2002.

[19] J. Le Boudec, D. McDonald, and J. Mundinger, "A generic mean field convergence result for systems of interacting objects," in *Proc. 4th Int. Conf. on the Quantitative Evaluation of SysTems (QEST 2007)*, Edinburgh, UK, 16–19 Sep. 2007, pp. 3–15.

[20] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*. Prentice-Hall, 1998.

[21] "The enabling grids for e-science project," online: http://www.eu-egee.org.

Benny Van Houdt received his M.Sc. degree in mathematics and computer science, and a Ph.D. in science from the University of Antwerp (Belgium) in July 1997, and May 2001, respectively. From October 2001 onwards he has been a postdoctoral fellow of the FWO-Flanders. In 2007, he became a professor at the Mathematics and Computer Science Department of the University of Antwerp. His main research interest goes to the performance evaluation and stochastic modelling of communication networks and random access systems in particular.

Chris Develder received the M.Sc. degree in computer science engineering and a Ph.D. in electrical engineering from Ghent University (Ghent, Belgium), in July 1999 and December 2003 respectively. From October 1999 to December 2003 he worked at the Department of Information Technology (INTEC), at Ghent University, mainly on optical packet/burst switched network paradigms. From January 2004 to September 2005, he worked for OPNET Technologies, to rejoin INTEC in October 2005. Since October 2007, he is professor at Ghent University, where his research interests include dimensioning, modeling and optimizing optical Grid networks and their control and management, as well as multimedia and home network software and technologies.

Juan F. Pérez received his M.Sc. in Industrial Engineering from Universidad de los Andes, Bogotá, Colombia in 2006. He is now member of the research group Performance Analysis of Telecommunication Systems (PATS) at the Mathematics and Computer Science Department of the University of Antwerp, Belgium. His main interests are around computational probability, stochastic modeling and performance evaluation of optical networks.

Mario Pickavet received an M.Sc. and Ph.D. degree in electrical engineering, specialized in telecommunications, from Ghent University in 1996 and 1999, respectively. Since 2000, he is professor at Ghent University where he is teaching courses on discrete mathematics, multimedia networks and network modeling. His current research interests are related to broadband communication networks (WDM, IP, (G-)MPLS, Ethernet, OPS, OBS) and include design, long-term planning, techno-economical analysis and energy efficiency of core and access networks. Special attention goes to Operations Research techniques that can be applied for routing and network design. In this context, he is currently involved in several European and national projects, such as the Network of Excellence "Building the Future Optical Network in Europe" (BONE), DICONET and ECODE.

Bart Dhoedt received a degree in Engineering from Ghent University in 1990 and became professor at the Faculty of Applied Sciences, Department of Information Technology. He is responsible for several courses on algorithms, programming and software development. His research interests are software engineering, distributed (autonomic) systems, grid and cloud computing, and thin client computing.