Mixing Malleable and Rigid Bandwidth Requests for Optimizing Network Provisioning

Sebastien Soudan and Pascale Vicat-Blanc Primet
INRIA, Université de Lyon, sebastien.soudan@ens-lyon.fr

***Abstract*—To address the anticipated sporadic terabyte demand generated by high-end time-constrained applications, dynamically reconfigurable optical networks services are envisioned. However, the time and rate granularities of a bandwidth reservation service and those of transfer tasks using the reserved capacity are not necessarily of the same order of magnitude. This may lead to a poor resource utilization and over-provisioning. This paper explores how request aggregation is able to limit this problem. Interactions between a bandwidth reservation service and a data mover service guaranteeing time-constrained data transfer are investigated. We formulate the underlying optimization problem and propose a linear-program-based strategy for bandwidth provisioning when malleable transfer requests are known in advance. Simulations show that the temporal parameters of requests (deadline and patience) are the dominant criteria and that a small malleability can improve performance a lot.**

I. Introduction

Large scale distributed scientific but also industrial data intensive applications require use of high performance network infrastructure. For example, collaborative engineers need to interact with massive amounts of data to analyze the simulation results over high resolution visualization screen fed by storage servers during time-limited working sessions. To address this anticipated sporadic terabyte demand, dynamically reconfigurable optical networks have been explored in labs and testbeds. One of the directions recently investigated in large scale distributed computing and data processing systems [1], [2], [3], [4] is the capacity of dynamically establishing dedicated lambda path. To develop and expand these facilities to an industrial context [5], new management and control functions are required to adapt existing Telecom network infrastructures to deliver commercial IT services for company customers. In currently proposed frameworks and interfaces [6], the users will express their bandwidth requirements to the network provider in terms of rate and deadline. A commercial interface will include the maximum cost users are willing to sustain. On the other side, the network providers will publish their services according to their policies and negotiations to maximize utilization rates. However, the granularities a realistic and efficient optical bandwidth reservation service can offer, typically in order of gigabits per seconde, are not flexible enough to meet the heterogeneity of end-users requirements. This may lead to a poor resource utilization, over-provisioning and unattractive service's pricing. We propose then to adapt the service to the real needs by providing a flexible interface and mixing different provisioning strategies. The key idea is to separate rigid requests from the elastic ones. We consider rigid requests (for real-time video and audio conferencing applications for example) concern deterministic bandwidth provisioning while malleable demands are for time-constrained huge data transfers. An intermediary service [7] is in charge of carrying out and grouping giant transfer tasks (with a volume greater than several GB) in specified time intervals. It aggregates the malleable requests and provision the bandwidth accordingly. Such a service considers that most users will care on transferring large amount of data in a limited and predictable time frame rather than requiring exactly a fixed rate for a given time-window. This approach relieves the end user from burdens of bandwidth reservation and allocation, while ensuring flow-completion in a timely fashion [8]. Users flexibly express their bandwidth provisioning requests with minimal rate, maximal deadline but also volume and maximal achievable rate. The goal of this paper is to demonstrate that the flexibility proposed increases the efficiency in lambda-path usage. In particular, we study how the network provisioning service, with the help of the data mover service, can optimally serve different traffic patterns distributions.

The remainder of this paper is organized as follows. Section II defines the model and formulates the problem. In Section III, simulation results are presented to demonstrate the impact of the homogeneity of the request's parameters such as volume, rate, patience and the influence of the proportion of malleable requests in the mix. Section IV contains a discussion on rate granularity and limitations of proposed approach. Related works are reviewed in Section V before concluding in Section VI.

II. Model and problem formulation

Let us consider a network model defined as a cloud, owned by a *network operator* (NO), exposing a *Dynamic Bandwidth Provisioning service* which provides on-demand lambda path or provisioned links between a set of end points and peering points with other network clouds. The network is defined by its set of points of presence $s \in \{e_1, ...e_S\}$ where S is the number of exposed points. We assume here that any two points exposed by one cloud can be source and destination of a reservation. On top of this bandwidth provisioning service, the *service provider* (SP) offers *users* (U) a service of scheduled and guaranteed data transfers from one end point to another and a service of bandwidth provisioning.

The SP that schedules users' requests tries to maximize the resource usage. As a different economic agents from the NOs, SPs do not have access to the routing plane and hence can only

manage movement capacity or bandwidth capacity between the network points of presence.

The three main actors in this model are then: the users that aim at transferring files from one site to an other with strict completion deadlines or renting fixed bandwidth for a time window, the service provider that tries to maximize the resource usage by scheduling users' requests and provisioning network paths. The network operator that provides the lambda path functionality.

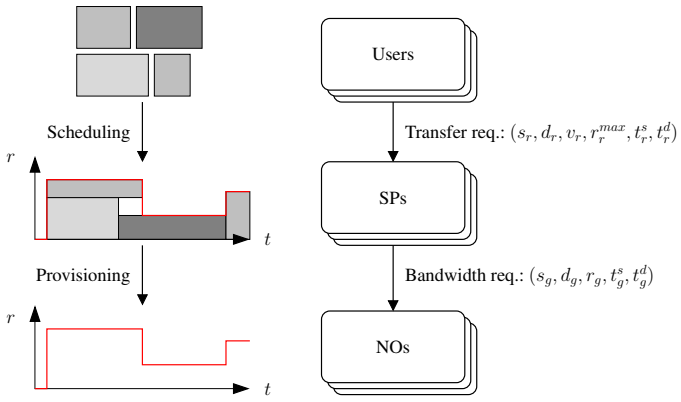

Fig. 1. Transfer and bandwidth requests exchanged between actors. SPs group users' requests in bandwidth requests issued to NO.

$(s_r, d_r, v_r, r_r^{max}, t_r^s, t_r^d)$	user to SP transfer request r
(s_r, d_r)	source-destination pair
v_r	volume
r_r^{max}	maximum rate
$[t_r^s, t_r^d]$	reservation window
$t \mapsto p_r(t)$	bandwidth allocation profile of r
r_r^{min}	minimum constant rate to serve r
P_r	patience of r
$g^{m,n} = (s_g, d_g, r_g, t_g^s, t_g^d)$	SP to NO bandwidth request issued on period m for period n
(s_g, d_g)	source-destination pair
r_g	requested bandwidth
$[t_g^s, t_g^d] = [n.\delta, (n+1).\delta]$	reservation window

TABLE I
REQUEST-RELATED NOTATIONS.

$[t_r^s, t_r^d]$	reservation window of request r
t_r^a	submission time of request r
δ	duration of a period
α	advance in time required to reserve/provision bandwidth
$[t_g^s, t_g^d] = [n.\delta, (n+1).\delta]$	reservation window of bandwidth request $g^{m,n}$
$t_g^a = m.\delta - \alpha$	submission time of bandwidth request $g^{m,n}$

TABLE II
TIME-RELATED NOTATIONS.

Fig. 1 shows the relations between actors and the requests format. Users are issuing transfer requests to SP which groups them and issues bandwidth requests to NO based on the bandwidth and time window requirements of each group.

A. Requests and constraints

The *transfer requests* sent by users to the SP are defined as follows. Notation are summarized in Table I and II . Each request r is a 6-uple $(s_r, d_r, v_r, r_r^{max}, t_r^s, t_r^d)$ where s_r is the source, d_r is the destination v_r is the volume to transfer, r_r^{max} is the maximum rate sender can send data. Transfer can only start after t_r^s and must be finished before t_r^d (reservation window). t_r^a is the arrival date of request r and t_r^r is the date of the request acceptance decision. Minimum constant rate r_r^{min} is defined as $r_r^{min} = \dfrac{v_r}{t_r^d - t_r^s}$ and *patience* P_r as $P_r = \dfrac{r_r^{max}}{r_r^{min}}$. The closer to 1 patience is the less malleable the request is. The request can't be served and is *invalid* if $P_r < 1$ due to constraint 1 which is defined thereafter.

Note that Users can also ask for fixed bandwidth by tailoring the request to the SP so that there is no patience ($P_r = 1$). A *bandwidth-specified user request* of rate r_r between t_r^s and t_r^d is thus specified as a 5-uple $(s_r, d_r, r_r, t_r^s, t_r^d)$ rewritten by the SP as the 6-uple: $(s_r, d_r, (t_r^d - t_r^s).r_r, r_r, t_r^s, t_r^d)$. Therefore SPs expose two different services with two request formats: one for malleable transfers, one for bandwidth-on-demand.

Now consider interactions between a SP and a NO. The *bandwidth requests* sent to NOs are not malleable and are defined as follows: A bandwidth request g is a 5-uple $(s_g, d_g, r_g, t_g^s, t_g^d)$ where s_g is the source, d_g is the destination, r_g is the requested rate, t_g^s is the start time and t_g^e the end of the reservation. Similarly to transfer requests, t_g^a is the arrival date of request g. We assume that bandwidth is provisioned α in advance, by slots of duration δ. This assumption is reasonable since NOs will have different SPs and time slots will mitigate the fragmentation. This also enables NOs to ensure the planned configuration of the network for next time slot is correct before applying to it.

Assumption 1: Bandwidth request issued for slot n is constrained to have: $t_g^a \leq t_g^s - \alpha$, $t_g^s = n.\delta$ and $t_g^e = (n+1).\delta$. In the remaining we assume a negotiation process that allow SPs to choose *a priori* the NO they want to use. Thus interaction between one given SP and one given NO are studied. Then if this request is issued by the SP to the NO at $t_g^a = m.\delta - \alpha$ with $m \leq n$, it will be noted $g^{m,n}$. Final bandwidth request for slot n is thus noted: $g^{n,n}$. In order to avoid over-estimation of in-advance bandwidth reservations, we assume (Assumption 2) that a SP can only re-provision for a given slot, by increasing the requested rate.

Assumption 2: At time $m^{'}.\delta - \alpha$, when updating advance bandwidth requests previously made at $m.\delta - \alpha$ for slot n, SP can only increase requested bandwidth. More formally: $\forall m < m^{'} \leq n, r_{g^{m,n}} \leq r_{g^{m',n}}$
This is illustrated on Fig. 2 where new bandwidth requests for slots n and $n+1$ issued at time $n.\delta - \alpha$ are shown in plain line while old bandwidth requests are dashed.

Assumption 3: SPs can only rent end to end bandwidth resource to network operator. SPs do not have routing facilities. As previously stated, Assumption 3 is based on realist situation in which network operators do not provide routers nor direct access to routing to their customers. This implies that there is

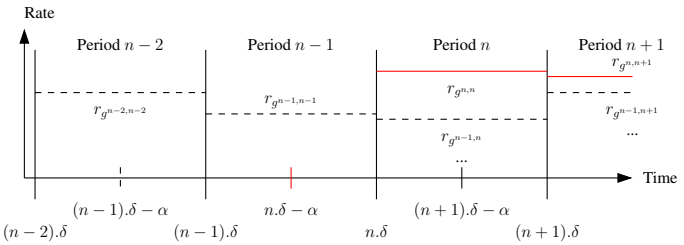

Fig. 2. Bandwidth provisioning slots $n-2$ to n seen at time $n.\delta-\alpha$.

no routing opportunity from the SP's point of view. Routing issues are addressed by the NOs. This also avoid the need to expose detailed topological information of the core network.

We define the *bandwidth allocation profile of* r as a step-function $t \mapsto p_r(t)$ defining the rate allocated to this transfer over time. A valid *bandwidth allocation profile* p_r must verify constraints 1, 2, 3.

$$\forall t \in [t_r^s, t_r^d], \quad 0 \leq p_r(t) \leq r_r^{max} \tag{1}$$

$$\forall t \notin [t_r^s, t_r^d], \quad p_r(t) = 0 \tag{2}$$

$$\int_{t_r^s}^{t_r^d} p_r(t)\ \mathrm{dt} = v_r. \tag{3}$$

a_r^s is the actual start time of transfer r and a_r^f its actual finish time. More formally, $a_r^s = \min\{t | p_r(t) \neq 0\}$ and $a_r^f = \max\{t | p_r(t) \neq 0\}$.

In this model, SPs do bandwidth reservations with same source and destination sites as transfer requests. Bandwidth reservation have to satisfy the following constrains to support transfer requests. Let's consider a set of N transfer requests $R = \{r_1, \ldots, r_N\}$ and a set of M non-overlapping[1] bandwidth requests $G = \{g_1, \ldots, g_M\}$; validity constraints are (in addition to 1, 2 and 3 for each requests in R) the following:

$$\forall g \in G, \forall t \in [t_g^s, t_g^e], \quad \sum_{r \in R} p_r(t) \leq r_g \tag{4}$$

$$\forall r \in R, \forall t \notin \bigcup_{g \in G} [t_g^s, t_g^e], \quad p_r(t) = 0 \tag{5}$$

$$\forall r \in R, \quad s_r = s_g \tag{6}$$

$$\forall r \in R, \quad d_r = d_g. \tag{7}$$

Fig. 3 shows a group of transfer requests and the corresponding bandwidth request to serve them. Due to Assumption 3, and without loss of generality, the remainder of this paper focuses on as transfer requests with same source/destination pair (sharing the same path) as transfer requests with different source or destination do not interact.

B. Requests state diagram

User's transfer or bandwidth-specified requests are received and processed by the SP. State diagram of requests comprises four states: (1)*New*, (2) *Scheduled*, (3) *Granted*, (4)*Rejected*.

A request r is: (1) *New* when the request has just been received and is valid but has been neither accepted nor rejected, (2) *Scheduled* when it has been accepted but allocated profile $t \mapsto p_r(t)$ can still be changed, (3) *Granted* when it can't be changed anymore (4) and finally *Rejected* when the request is not accepted by the SP. Theses states and allowed transitions are depicted in Fig. 4.

[1] $\forall g, g' \in G, (t_g^s, t_g^e) \cap (t_{g'}^s, t_{g'}^e) = \emptyset$.

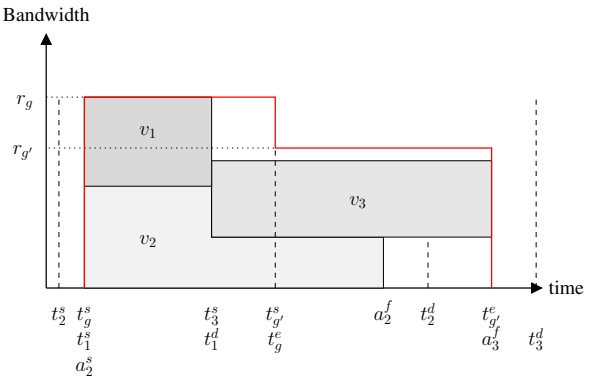

Fig. 3. Transfer requests ($r \in \{1, 2, 3\}$) grouped in bandwidth requests g and g'.

The transitions from *New* to *Scheduled* or *Rejected* depend on the decision taken by the SP when first considering this request. The state of request r is changed from *Scheduled* to *Granted* t_{grant} before t_r^s in order to give some time to the sender before transfer's start time. Since scheduling is done every time slots, this transition is implemented between lines 4 and 10 of Algorithm 1 as allocations of about-to-be-granted requests can not be changed later.

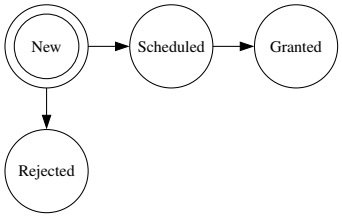

Fig. 4. Transfer requests' state diagram: once *Scheduled* a transfer can't be rejected anymore but profile can still change until transfer reaches *Granted* state.

C. Problem formulation

For every slot n at $n.\delta-\alpha$, the SP has to decide for each source/destination pairs which bandwidth request to issue for next slot to accommodate already known transfer requests R_n ($\forall r \in R_n, t_r^a \leq n.\delta-\alpha$) and how to schedule these transfers. We first formulate the problem under the non-rejection assumption. In this context, the objective is to minimize the aggregated provisioned capacity. If we take into account the rejection, the problem becomes a maximization of a global reward. We discuss this extension in section .

According to previously defined state diagram, R_n is partitioned into three subsets $R_n = R_n^{new} \bigcup R_n^{sched.} \bigcup R_n^{granted}$ where $R_n^{granted}$ is the set of requests already granted during slots before $n.\delta-\alpha$, R_n^{new} contains new valid requests which have not yet been scheduled (or rejected) and $R_n^{sched.}$ contains

R_n	set of all user requests know at $n.\delta - \alpha$
R_n^{new}	requests of R_n that have not yet been scheduled (received between $(n-1).\delta - \alpha$ and $n.\delta - \alpha$)
$R_n^{sched.}$	already scheduled requests
$R_n^{granted}$	requests with profile that can't be changed anymore
G_n^{final}	bandwidth requests that can't be changed anymore
$G_n^{prev.}$	bandwidth requests for upcoming slots as requests at $(n-1).\delta - \alpha$
G_n^{new}	bandwidth requests for upcoming slots as sent to NO at $n.\delta - \alpha$

TABLE III
STATE-RELATED NOTATIONS.

transfer requests that can still be re-scheduled. It can be noted that requests in $R_n^{sched.} \bigcup R_n^{granted}$ were already in R_{n-1}. Notations are summarized in Table III.

Let $G_n^{final} = \{g^{i,i} | 1 \leq i \leq n-1\}$ be the set of bandwidth reservation made for slot up to $n-1$ (they can't be modified anymore), $G_n^{prev.} = \{g^{n-1,i} | n \leq i \leq M\}$ and $G_n^{new} = \{g^{n,i} | n \leq i \leq M\}$ where M is the greatest slot utilized by a transfer request.

Using previously defined validity constraints for bandwidth requests and transfer requests and constraints on increasing bandwidth requests, we formulate the problem as:

$$
\begin{aligned}
BP(n): \quad \text{minimize:} \quad & \sum_{g \in G_n^{final} \bigcup G_n^{new}} r_g \\
\text{subject to:} & \\
\forall r \in R_n, \forall t \in [t_r^s, t_r^d], \quad & 0 \leq p_r(t) \leq r_r^{max} \\
\forall r \in R_n, \forall t \notin [t_r^s, t_r^d], \quad & p_r(t) = 0 \\
\forall r \in R_n, \quad & \int_{t_r^s}^{t_r^d} p_r(t) \ \mathrm{dt} = v_r \\
\forall g \in G_n^{final} \bigcup G_n^{new}, \forall t \in [t_g^s, t_g^e], \quad & \sum_{r \in R_n} p_r(t) \leq r_g \\
\forall t \notin \Big(\bigcup_{g \in G_n^{final} \bigcup G_n^{new}} [t_g^s, t_g^e] \Big), \forall r \in R_n, \quad & p_r(t) = 0 \\
\forall i, n \leq i \leq M, \quad & r_{g^{n-1,i}} \leq r_{g^{n,i}}
\end{aligned}
$$

First constraint of this linear program states that service rate must lie within the maximum sending rate constraint given by the user. Second constraints service rate to be null out of the reservation window while third ensures sum of the service rate equals requested volume. Fourth constraint says that the bandwidth provisioned must be higher than the sum of the profiles of served transfer requests and fifth constraint ensures transfer requests to be served with a bandwidth request. The last constraint says that from one negotiation to another, bandwidth requested by SPs to NOs can't decrease.

Let $\mathcal{I}$ be the set of time intervals defined by dividing the time axis on all t_g^s, t_g^e, t_r^s and t_r^d. In this case, $\gamma_{r,i}$ will be 1 if request r can be served on interval i and 0 else, $\gamma_{g,i}$) equals 1 if bandwidth request g covers interval i and 0 else (all i are supposed to be covered by a bandwidth request g possibly with $r_g = 0$), l_i is the length of interval i and $p_{r,i}$ the constant rate of $p_r(t)$ on interval i. Then the problem BP(n) can be rewritten as a linear program:

$$
\begin{aligned}
BPLP(n): \quad \text{minimize:} \quad & \sum_{g \in G_n^{final} \bigcup G_n^{new}} r_g \\
\text{subject to:} & \\
\forall r \in R_n, \forall i \in \mathcal{I}, \quad & 0 \leq p_{r,i} \leq r_r^{max} \\
\forall r \in R_n, \forall i \in \mathcal{I}, \quad & (1 - \gamma_{r,i}).p_{r,i} = 0 \\
\forall r \in R_n, \quad & \sum_{i \in \mathcal{I}} \gamma_{r,i}.p_{r,i}.l_i = v_r \\
\forall i \in \mathcal{I}, \forall g \in G_n^{final} \bigcup G_n^{new}, \quad & \gamma_{g,i}. \sum_{r \in R_n} p_{r,i} \leq r_g \\
\forall j, n \leq j \leq M, \quad & r_{g^{n-1,j}} \leq r_{g^{n,j}}
\end{aligned}
$$

where variables are: $\{p_{r,i} | r \in R_n^{new} \bigcup R_n^{sched.}, i \in \mathcal{I}\}$ and $\{r_g | g \in G_n^{new}\}$. $\{p_{r,i} | r \in R_n^{granted}, i \in \mathcal{I}\}$ is not part of the variable as *Granted* requests' profiles can't be changed anymore.

It can be proved that provided requests in R_n^{new} are *in advance* requests, meaning they have their start time t_r^s after $n.\delta$, BPLP(n) has a solution. Requests of R_n^{new} with t_r^s before $n.\delta$ are called *immediate* requests. To do so we prove by induction the solution space is not empty. Let assume BPLP($n-1$) has a solution. (1) All *New* requests are valid requests and thus have $P_r \geq 1$ meaning that a pure rectangle of r_r^{max} on $[t_r^s, t_r^d]$ can be used as their bandwidth allocation profiles. (2) To serve these *New* requests starting from solution of BPLP($n-1$) only requires to increase bandwidth of slot greater than n which is allowed by assumption 2 and reusing same profile as generated by previous run for requests in $R_n^{sched.} \bigcup R_n^{granted}$. This demonstrates that BPLP(n) has at least one solution without changing profile of *Granted* requests. First iteration BPLP(1) can obviously be solved.

We can observe in the formulation of BPLP(n) that requests in $R_n^{granted}$ could have their profiles $p_r(t)$ be summed and processed as a single profile as they won't be changed and request-centric constraints (1-3) have been verified in BPLP($n-1$) for these requests. This would allow to forget past history of the allocations and prevents problem from growing at every iteration.

Once this problem has been solved, *New* and *Scheduled* requests are marked as *Scheduled* or *Granted* depending on their start time. $R_{n+1}^{granted}$ and $R_{n+1}^{sched.}$ can thus be prepared for next slot while R_{n+1}^{new} is filled when requests arrive. Once new bandwidth reservation requests have been sent to NO set G_{n+1}^{final} and $G_{n+1}^{prev.}$ can be determined . The whole procedure for provisioning and scheduling transfers is depicted in Algorithm 1. It takes the requests received so far with there states and the previous bandwidth reservations. It outputs requests with updated states and profiles, and new bandwidth reservations.

D. Complexity analysis

This problem can be seen as a min-cost multicommodity flow over time with uniform path-lengths problem on a simple dumbbell graph. Cost function would sum the difference of provisioned bandwidth (max of instant bandwidth) minus bandwidth already allocated transfers of $R_n^{granted}$. Then the graph considered is made of input edges between the vertex representing the request and the input vertex of the central edge. Transit time of this input edge equals the start time of the request and similarly an output edges of transit time equals to the difference between the largest deadline minus the one of the transfer request connects the output vertex of central edge and a sink representing the request. The central edge of the dumbbell has no transit time. The schedule of flows on the central edge is the schedule of transfer requests. This problem has been proved to be solvable in polynomial time in Theorem 1 of [9]. Then, the problem enjoying LP formulations works in the time proportional to the number of optimization variables. Given R the total number of requests considered per schedule period and G the number of period considered, there is in about at most $G+R(2(R+G)-1)$ optimisation variables ($p_{r,i}$) as the number of intervals defined by start time and deadline and period is less than $2(R+G)-1$. The linear program has $6R(R+G)+3G-1$ constaints. Using the complexity of Karmarkar's algorithm for a fixed maximum size of variable, we obtain a complexity in $O((G+R(2(R+G)-1))^{3.5})$. Another solution to solve this is to use a static min-cost multicommodity flow computation on an time-expanded networks of $O((2R+2)^2)$ nodes and $O((2R+2)(2R-1))$ links as proposed by Theorem 2 of [9]. If bandwidth allocation profile is restricted to constant-value single-interval form, the problem would have been NP-Complete as demonstrated in [7]. Therefore, allowing rational step functions and malleable requests is necessary to be able to minimize the provisioning cost.

III. Performance evaluation

In the previous section we have unified malleable transfer requests and fixed bandwidth requests in a common format. We have formulated the Dynamic Bandwidth Provisioning problem BPLP. Then we have proposed and proved a linear program based solution for any given set of malleable and fixed transfer requests. The goal of the performance evaluation is to explore how time and rate granularities as well as requests malleability impact resource provisioning and utilization. In order to evaluate the impact of requests' characteristics, we performed simulations with different workloads. The proposed provisioning algorithm has been implemented in jBDTS [10] which is used for the simulations.

Rates proposed by NO are generally discrete. Therefore, in the following simulations Algorithm 1 is compared to a modified version of it which uses BPLP as a linear relaxation of the mixed integer linear problem with discrete value for r_g. In this modified version, rounding of r_g to the upper discrete value is performed just before sending connectivity requests at line 11 of Algorithm 1. We used discrete steps of 100 Mbps.

Algorithm 1 Schedule and provision at $t = n.\delta - \alpha$

Input: $R_n^{new}, R_n^{sched.}, R_n^{granted}, G_n^{final}, G_n^{prev.}$
Output: $R_{n+1}^{sched.}, R_{n+1}^{granted}, G_{n+1}^{final}, G_{n+1}^{prev.}, \{t \mapsto p_r(t) | r \in R_{n+1}^{sched.} \bigcup R_{n+1}^{granted}\}$

// Initialize set of req. for next slot
1: $R_{n+1}^{sched.} \leftarrow \emptyset$
2: $R_{n+1}^{granted} \leftarrow R_n^{granted}$
// Determine profiles for transfer req. and bandwidth req.
3: Solve BPLP(n)
// Update transfer req.'s states
4: **for all** $r \in R_n^{new} \bigcup R_n^{sched.}$ **do**
5: **if** $t_r^s - t_{granted} \leq (n+1).\delta - \alpha$ **then**
6: $R_{n+1}^{granted} \leftarrow R_{n+1}^{granted} \bigcup \{r\}$
7: **else**
8: $R_{n+1}^{sched.} \leftarrow R_{n+1}^{sched.} \bigcup \{r\}$
9: **end if**
10: **end for**
// Send bandwidth req. to NO
11: Issue bandwidth requests $g \in \{G_n^{new}\}$ to NO.
// Update bandwidth req. sets for next slot.
12: $G_{n+1}^{final} \leftarrow G_n^{final} \bigcup \{g^{n,n}\}$
13: $G_{n+1}^{prev.} \leftarrow G_n^{new} \backslash \{g^{n,n}\}$
14: **return** $R_{n+1}^{sched.}, R_{n+1}^{granted}, G_{n+1}^{final}, G_{n+1}^{prev.}, \{t \mapsto p_r(t) | r \in R_{n+1}^{sched.} \bigcup R_{n+1}^{granted}\}$

All two first experiments use a common set of default parameters parametrized by the slot duration δ:

- Slot duration: δ;
- Provisioning advance: $\alpha = \delta/2$;
- Granting advance: $t_{grant} = \delta/20$;
- Transfer request inter-arrival: $\delta/12$;
- Number of requests: 2000;
- Requests' attributes:
  - $t_r^d - t_r^s = d = \delta/2$;
  - $t_r^s - t_r^a = 2.\delta$;
  - $r_r^{min} = R = 53$ Mbps ($v_r = R.\delta/2$);
  - $P_r = 2$.

Preliminary experiments have shown that increasing volume heterogeneity has no impact which can be explained by important overlapping of reservation windows as large requests encompass small ones and our assumption of unbounded provisionable capacity.

A. Impact of reservation window duration heterogeneity

In this experiment we mix requests with same volume but heterogeneous reservation windows. We alternatively submit one request with duration $h_d.\delta/2$ and one with $(2-h_d).\delta/2$.

Fig. 5 shows that duration heterogeneity has a very important influence on the unused capacity. This is because decreasing the duration of some requests without varying their volumes, increases their r_r^{min}. With as small h_d as $1/30$, r_r^{min} is 30 times higher than in default case. Furthermore, due to no-rejection policy, reserved bandwidth has to be greater than r_r^{min} even when constraining requests have small

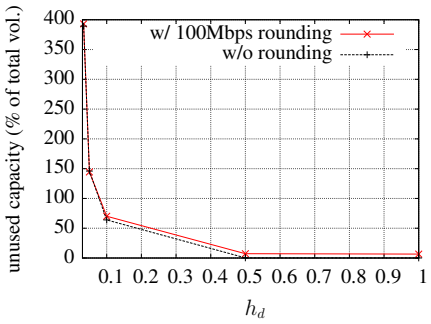

Fig. 5. Provisioned but unused bandwidth as a function of transfer duration homogeneity parameter h_d without and with rounding to the upper 100Mbps.

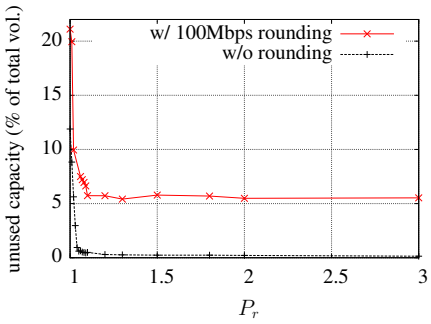

Fig. 6. Provisioned but unused bandwidth as a function of patience without and with rounding to the upper 100Mbps.

volumes compared to the volume they force SP to provision. We can observe that performance of discrete and continuous bandwidth cases are similar with the 5% difference previously observed.

B. Impact of patience

Given that most of the time the duration and the volume of request will be fixed by the application semantic, we now observe the impact of the patience factor, which represents the request's malleability. In our model, patience is higher or equal to 1 otherwise requests wouldn't be accepted. If patience equals 1, requests can't be reshaped and have to be scheduled as one single rectangle at r_r^{max} during $[t_r^s, t_r^d]$. The impact of this factor can observed on Fig. 6 where extra reserved bandwidth decreases quickly as P_r increases. When P_r is greater than 1.5, performance is constant. This means that the patience factor has a strong impact.

C. Impact of proportion of malleable requests

In this set of experiment, we used two classes of requests: one with patience equal to 1 (rigid requests) and one with varying patience (malleable requests).

Requests used here have the following parameters: There is 30 clients each acting as an ON-OFF source with exponential OFF time of mean 2 hours. ON times are given by the reservation windows $[t_r^s, t_r^e]$. Each client submits 8 requests. The volumes of each request are derived from a Pareto distribution with minimum volume equals to 100GB

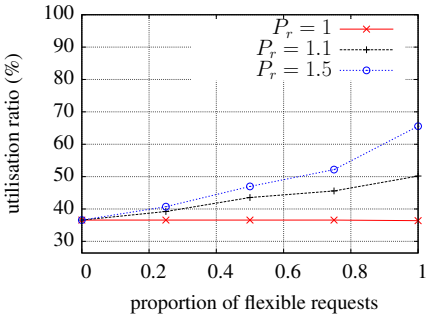

Fig. 7. Provisioned but unused bandwidth as a function of p for different patiences ($P_r \in \{1, 1.1, 1.5\}$) with rounding to the upper 100Mbps.

and shape parameter equals to 1.5 leading to an average volume of 300GB per request. Each source has a maximum sending rate of 100Mbps or 1Gbps randomly chosen with equal probability. Similarly each transfer can be flexible and have a varying patience ($P_r \in \{1, 1.1, 1.5\}$) or can have no flexibility ($P_r = 1$), probability of being a malleable requests is p. Slot duration is equal to 24 hours. As an example, with a sending rate of 100Mbps, a 300GB transfer has a minimum completion time (with patience equals to 1) of 6.67 hours. In this experiment, as maximum rate is fixed, increasing patience decreases minimum rate r_r^{min} and thus increases transfer's potential duration.

We can observe on Fig. 7 that provisioning of this considered set of requests lead to an important waste of bandwidth when there is no flexibility ($P_r = 1$ or $p = 0$). But gain is linear with proportion of malleable requests as r_r^{min} of this kind of requests is P_r times smaller when the requests are malleable and the number of such requests is in proportion p of the total number of requests.

The second observation is that increasing the patience is an efficient way to decrease the over-provisioning. Increasing the patience of 10% from $P_r = 1$ to $P_r = 1.1$, increases the utilization ratio from 36% to 50% when all requests are flexible. However this gain is not linear with patience as going from $P_r = 1.1$ to $P_r = 1.5$ only improves the utilization ratio by 17%.

IV. Discussion

While users' requests can be seen as flow of packets: FTP transfers over a TCP connection for example, and thus have a great flexibility in term of reachable rates, provisioned bandwidth has to be considered as circuits with guaranteed bandwidth. These circuits can be realized through MPLS circuits and thus offer fine granularity in term of rate. In this case, rational solution of the linear program can be well approximated by the provisioned rate. But they can also be implemented using coarse grained technologies such as MEF ev-lines[11] which offer Ethernet rates: 10, 100, 1000, 10000Mbps with the possibility of trunking them. In this case, the problem would require to solve a mixed integer linear program which is known to be much more complex. Instead, in the proposed solution, a linear program is solved

and rate is rounded towards the next greater available rate. But as the optimization problem is solved at every periods and requests overlapping several periods, achieved solution is better than a simple rounding: at next schedule, some requests have the opportunity to be moved in order to fully-utilize already reserved bandwidth and thus prevent increase of provisioned bandwidth in other time slots. This migration is due to the minimization of the objective function. When the malleable transfer requests come continuously and sufficiently in advance (more than one period) this solution is efficient. This user behavior can be favored by economical or qualitative incentives such as different pricing as proposed by airliners or such as different probability of being rejected as first-come-first-served admission rules would do. It is also interesting to consider other objective functions, for example, the total provision cost, Roughly speaking, the scheduling optimization problems with different objective functions can be treated using the same techniques as in this paper, if the objective function can be integrated into the constrained optimization problem in a linear form. Another point worth noting is the work done in [12] on the determination of optimal service levels that a network operator can propose to a client so that he can minimizes the unused capacity in average. This result requires knowledge of the distribution of provisioned rate to distribute the service levels.

V. Related works

The dynamic provisioning service supposedly offered by NOs to SPs and to some extent by SPs to Users is actively studied. Current state-of-the-art is that these circuit services are configured manually either by fax, phone, e-mails or in selected cases with first generation web services.

This kind of service provides the capability of dynamically establishing dedicated connection oriented circuit switching or lambda path like [1], [2], [13]. As an example of these new services, UCLP (User controlled Light Paths) allows users and applications to partition and configure the resources of an optical network. CHEETAH project [14] (Circuit-switched High-speed End-to-End Transport Architecture) is also towards end-to-end connection oriented circuit in a GigaEth over SONET transport network with GMPLS control functions at the network elements. The goal of the DRAGON project [15] (Dynamic Resource Allocation via GMPLS Optical Networks) is to develop functions such as Network-Aware Resource Broker (NARB) and Application-Specific Topology Builder (ASTB) required by the network infrastructure for performing immediate and in-advance reservations of networks resources for connections in a heterogeneous and multi-domain transport network.

[16] suggests to make network reconfigurations available to application users by making visible all the resource and allowing them to send signaling message in carriers' networks. This approach rises some confidentiality concerns but can be applied to specific distributed applications like e-science deployed over National Research and Educational Networks (NREN).

To develop and adapt these facilities into an industrial context, CARRIOCAS project [17] proposes new management and control functions to evolve existing Telecom network infrastructures to deliver commercial services for company customers. Commercial services have to be built on an abstracted view of the infrastructures and resource signaling has to go through policy and business procedures before triggering network reconfiguration or resource reservations. Most of these current hybrid network implementations, connection services are permanently or semi permanently configured and managed by proprietary Network Resource Provisioning System (NRPS). Comparable service oriented approaches are also explored in [18], [19].

Several works generalize the dynamic provisioning approach to IT resources. In [20] two different integrations of service plane and network control plane are proposed. First one is Grid enabled GMPLS (G^2MPLS) which implements GMPLS control functions and extend them to include non-network resources as new switching capabilities. Second is a Session Initiation Protocol (SIP) over an Optical Burst Switching network (SIP-based OBS network) where SIP is used as service level signaling protocol and SIP proxies are used to access the OBS transport plane. They both suppose that network infrastructure topology, its capability and its states are accessible to the end-users. CARRIOCAS implements the virtual infrastructure concept by decoupling the management of the physical infrastructure (network element inventory, equipment and network maintenance and administration, performance monitoring, etc.) from the management of the services that can be delivered to external service providers.

A dynamic bandwidth scheduling scheme which exploits the quasi-flexible nature of connectivity reservations and considers as we do two broad classes of generated network traffic, streaming and elastic, with same type of QoS requirements has been proposed in [21]. This problem was also studied by Burchard in [22], where the concepts of malleable reservations to address bandwidth fragmentation was proposed. Our approach is similar to both proposals. However unlike in these related works, this paper formulates the problem and proposes linear programming approach which offers a tractable solution.

In previous works [7], [23], multi-step allocation of time-constrained transfer requests have been studied under fixed capacity constraints. This work extends and demonstrates how the approach applies to any in-advance bandwidth reservation service.

Other works looked at allocation of lambda-path to route demands but without considering the malleability and the in-advance reservation as in [24], [25]. [26] considers allocation and routing of non-malleable requests using branch and bound techniques.

In the context of self-sizing networks adaptively dimensioned as traffic changes several dynamic bandwidth allocation strategies have been proposed [27], [28], [29]. These approaches are all based on predictors. This paper considers a large fraction of the traffic is known in advance and is malleable. The bandwidth provisioning service is exposed to

users. Our solution benefits from traffic malleability and in-advance knowledge to adapt the traffic and to provision the network accordingly.

VI. Conclusion

This paper proposes a flexible interface to specify malleable transfer requests and fixed bandwidth requests in a common format. Then the Bandwidth Provisioning problem has been formulated. It provides a solution for *in advance* malleable and fixed requests scheduling and the provisioning of underlying network infrastructure in case this infrastructure has infinite resources (or supposed so). The proposed objective function is to minimize total volume of bandwidth reserved which make it a reasonable objective for service providers which will have to pay for this resource. Performance evaluation demonstrates how time and rate granularities as well as requests malleability affect resource provisioning and utilization.

Due to the accept-all policy, we showed that some transfer requests patterns lead to highly inefficient provisioning and large amount of wasted bandwidth. This exhibits a limit of the proposed strategy which would require to constrain users to specify requests within a given range of parameters to force the patience to a certain value, with for example $P_r \geq 1.5$, to avoid strict requests. Alternatively these requests with disproportionate r_r^{min} could also be charged differently based on their flexibility.

In future works, we plan to consider *immediate* transfer requests. These transfers requests can be served in the extra bandwidth allocated by BPLP or by an extra amount of bandwidth estimated by a traffic predictor focusing on *immediate* transfer requests. Finally another future work is to study the incentives that can lead users to be patient.

Acknowledgment

This work has been funded by the French ministry of Education and Research, INRIA, and CNRS, via ACI GRID's Grid'5000 project and Aladdin ADT and the CARRIOCAS project (pôle Systém@tic IdF).

References

[1] "G-lambda project website," Feb 2009, `http://www.g-lambda.net`.

[2] "UCLP: User Controlled Lightpaths website," Feb. 200b, `http://www.uclp.ca`.

[3] "Phoshorus project website," Feb. 2009, `http://www.ist-phosphorus.eu/`.

[4] "Enlightened Computing: Highly-dynamic Applications Driving Adaptive Grid Resources project website," Feb. 2009, `http://enlightenedcomputing.org/`.

[5] O. Audouin, D.Erasme, M. Jouvin, O. Leclerc, C. Mouton, P. Primet, D. Rodrigues, and L. Thual, "CARRIOCAS project: An experimental high bit rate optical network for for computing intensive distributed applications," in *BroadBand Europe'07*, Dec. 2007.

[6] "NSI-wg: OGF Network Service Interface WG (NSI-WG) website," Feb. 2009, `http://www.ggf.org/gf/group_info/view.php?group=nsi-wg`.

[7] B. Chen and P. Vicat-Blanc Primet, "Scheduling deadline-constrained bulk data transfers to minimize network congestion," in *CCGRID '07: Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*. IEEE Computer Society, May 2007, pp. 410–417.

[8] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 59–62, 2006.

[9] A. Hall, S. Hippler, and M. Skutella, "Multicommodity flows over time: Efficient algorithms and complexity," *Theor. Comput. Sci.*, vol. 379, no. 3, pp. 387–404, 2007.

[10] INRIA RESO team, "jBDTS: Bulk Data Transfer Scheduling service website," Feb. 2009, `http://www.ens-lyon.fr/LIP/RESO/Software.html`.

[11] "MEF: Metro Ethernet Forum website," Feb. 2009, `http://metroethernetforum.org`.

[12] G. N. Rouskas and L. E. Jackson, "Optimal granularity of mpls tunnels," in *In Proceedings of the 18th International Teletraffic Congress (ITC-18*. Elsevier Science, 2003, pp. 1–10.

[13] S. Figuerola, N. Ciulli, M. de Leenheer, Y. Demchenko, W. Ziegler, and A. Binczewski, "Phosphorus: single-step on-demand services across multi-domain networks for e-science," J. Wang, G.-K. Chang, Y. Itaya, and H. Zech, Eds., vol. 6784, no. 1. SPIE, 2007, p. 67842X. [Online]. Available: http://link.aip.org/link/?PSI/6784/67842X/1

[14] X. Zheng, M. Veeraraghavan, N. Rao, Q. Wu, and M. Zhu, "Cheetah: circuit-switched high-speed end-to-end transport architecture testbed," *Communications Magazine, IEEE*, vol. 43, no. 8, pp. s11–s17, Aug. 2005.

[15] T. Lehman, J. Sobiesky, and B. Jabbari, "Dragon: A framework for service provisioning in heterogeneous grid networks," *IEEE Communications Magazine*, vol. 44, no. 3, pp. 84–90, March 2006.

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

[17] "CARRIOCAS project website," Feb. 2009, `http://www.carriocas.org`.

[18] "Multi-Technology Operations Systems Interface (MTOSI) 2.0, TMF Forum," November 2008, http://www.tmforum.org/browse.aspx?catid=2319.

[19] F. Verdi, M. Magalhães, E. Cardozo, E. Madeira, and A. Welin, "A service oriented architecture-based approach for interdomain optical network services," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 141–170, 2007.

[20] 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, June 2008.

[21] S. Naiksatam and S. Figueira, "Elastic reservations for efficient bandwidth utilization in lambdagrids," *Future Gener. Comput. Syst.*, vol. 23, no. 1, pp. 1–22, 2007.

[22] L.-O. Burchard, "Networks with advance reservations: Applications, architecture, and performance," *J. Netw. Syst. Manage.*, vol. 13, no. 4, pp. 429–449, 2005.

[23] S. Soudan, B. Chen, and P. Vicat-Blanc Primet, "Flow scheduling and endpoint rate control in gridnetworks," *Future Generation Computer Systems*, 2008, to appear.

[24] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *Networking, IEEE/ACM Transactions on*, vol. 8, no. 5, pp. 598–607, Oct 2000.

[25] A. Gençata and B. Mukherjee, "Virtual-topology adaptation for wdm mesh networks under dynamic traffic," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 236–247, 2003.

[26] J. Kuri, S. Member, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville, "Routing and wavelength assignment of scheduled lightpath demands," in *in Procs. of ICOCN 2002, (Singapore*, 2003, pp. 1231–1240.

[27] Z. Sahinoglu and S. Tekinay, "A novel adaptive bandwidth allocation: wavelet-decomposed signal energy approach," *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 4, pp. 2253–2257 vol.4, 2001.

[28] Y. Liang and M. Han, "Dynamic bandwidth allocation based on online traffic prediction for real-time mpeg-4 video streams," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 51–51, 2007.

[29] B. Krithikaivasan, Y. Zeng, K. Deka, and D. Medhi, "ARCH-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic," *Networking, IEEE/ACM Transactions on*, vol. 15, no. 3, pp. 683–696, June 2007.