

A Subscription Model for Time-Scheduled Data Transfers

Dominik Klein, Michael Menth*, Rastin Pries,
Phuoc Tran-Gia
University of Würzburg, Institute of Computer Science
Email: {dominik.klein,menth,pries,trangia}
@informatik.uni-wuerzburg.de

Michael Scharf, Michael Söllner
Alcatel-Lucent Bell Labs Germany
Email: {michael.scharf,michael.soellner}
@alcatel-lucent.com

Abstract—We recently witness new services that can afford some delay until data transmission starts, but then benefit from a very large available bandwidth. A popular example is the migration of virtual machines between different sites of a geographically dispersed service provider.

In this paper, we propose a subscription model for time-scheduled data transfers. Transmission requests are served consecutively, giving the flows access to the physical bandwidth. This is in contrast to today's Internet where flows are served in parallel so that they compete for the available bandwidth. We present the architecture to enable such data beams. Furthermore, we model and analyze the performance under different conditions and compare it with concurrent transmission.

I. INTRODUCTION

The current Internet has evolved during the last years to a global provider of various new services and applications. The requirements of these new services are also evolving and differ from those of classical applications. Some of these applications can tolerate certain delay until the data transmission starts but then require a high data rate for the actual transmission. An example is the live migration of virtual machines between different locations of a data center. An important characteristic of virtual machine migration is that the total data transfer volume increases with increasing transfer time. The reason is that already copied memory pages need to be resent when they are overwritten by the virtual machine during the migration [1].

In this paper, we focus on reducing the time for such a migration. Transmission requests are served consecutively and have access to the physical transmission capacity if needed. A scheduler organizes the transmission of flows in a network. To avoid excessive waiting times for flows until they may start transmission, we introduce a subscription model that gives customers priority access to the scheduler so that a bounded amount of their transmission requests are preferentially served. We call this new network service with the subscription model data beaming, because transmission requests are served at very high bandwidth during a short time slot.

Another contribution is a performance comparison between our subscription model and concurrent transmission in today's Internet. With data beaming, flows need to wait for their transmission but are then transmitted very quickly. With

concurrent transmission, flows are transmitted immediately but it takes longer as multiple flows compete for the same resources, e.g., using TCP. We compare both options under different traffic characteristics. We further show that flows are fast transmitted if their customers do not exceed the bounds of their subscription model as the scheduler and the subscription model largely isolate different customers.

The rest of the paper is structured as follows. In Section II, we explain the motivation for live migration of virtual servers, technical basics, and its requirements for data transmission. Section III presents data beaming. In Section IV, we model data beaming and concurrent transmission and present a performance comparison based on analytical and simulative results in Section V. Section VI describes related work and explains how it differs from our work and finally, Section VII concludes this work.

II. COMMUNICATION REQUIREMENTS FOR LIVE SERVER MIGRATION

During the live migration of a virtual server, the virtual server process is moved from one machine to another while maintaining its operation most of the time. The processes may be moved only locally within a site or between remote sites of a service provider.

A. Motivation

Live migration facilitates the management and administration of virtual nodes and reduces downtime resulting from maintenance operations. Live migration enables load balancing and live workload mobility for a service provider without perceivable interruption of its provided services by currently connected clients. In addition, it improves energy efficiency of a service provider hosting several servers. To avoid idle or less loaded servers which nevertheless require about 66 percent of their total energy consumption [2], less loaded server processes may be migrated to only a few physical nodes so that the remaining nodes can be switched off to save energy.

B. Technical Principles

If virtual servers are migrated, usually only the working memory needs to be transferred. This is different when virtual servers are migrated between data centers that cannot share a common file system. Then, the local persistent data of the virtual server also needs to be transferred which significantly increases the data volume for the migration process [3].

This work was conducted within the Internet Research Center (IRC, currently being founded) at the University of Würzburg. The authors alone are responsible for the content of the paper.

* The author is now with the University of Tübingen.

An efficient approach for live migration is the pre-copy migration scheme [4]. During an initial push phase, the memory is sent in several iterations and modified pages of already sent parts of the memory are resent in subsequent iterations. Once a sufficiently large part of the memory has been transferred, the server on the source node is stopped and the remaining parts of the memory are sent to the destination node. Eventually, the server on the destination node is started.

Thus, a certain downtime during the migration process cannot be avoided, but it should be kept small. The length of the interruption mainly depends on the available bandwidth, the workload of the migrated server process [5], and the used technique to migrate a virtual node [4].

C. Requirements for Data Transmission

During the push phase of a migration process, the server on the source node keeps working and may modify already copied parts of the memory so that they need to be resent to the destination node. The rate at which already copied memory is overwritten by the server on the source node is called “dirtying rate”. The dirtying rate depends on the workload of the service offered by the virtual server.

Live server migration between different sites benefits from a high bandwidth for two reasons. First, more bandwidth reduces the number of copy iterations during the push phase and thus the transferred data volume. Second, with higher bandwidth, the remaining amount of data is also faster transferred in the stop-and-copy phase. In summary, the larger the bandwidth between the source and destination site, the shorter is the downtime of a migrated service. Therefore, high bandwidth is a key requirement for live migration.

The live migration of a single common web server for example consumes up to 500 Mbit/s during the last stop-and-copy phase [4]. If an upcoming maintenance operation requires the migration of several virtual servers within a couple of minutes, the necessary bandwidth between source and destination site may easily exceed several Gbit/s.

As live server migration is a planned process of the data center management, a time scheduling is possible, i.e., the migration could be started when sufficiently high bandwidth is available. Postponing the process is typically also possible, at least for a short time duration. This is a motivation for the data beaming concept that provides access to very high bandwidth, but possibly with some delay.

III. DATA BEAMING

We give an overview of data beaming, explain its architecture, and propose a simple scheduler that we use in our performance study.

A. Overview

A “data beam” is a transmission request that should be served at very high bandwidth so that data are transmitted within very short time. Customers wishing to transmit a certain data volume at a certain rate indicate that through the data beaming interface to the management of a transport network

which then returns a connection over which the customer can transmit the data. We call this service “data beaming”. It may be implemented in packet-switched or circuit-switched networks where GMPLS is an example as it is able to set up paths on a short timescale [6]. The authors of [7] for example present extensions to the GMPLS suite to support reservation of network resources in advance. Data beaming can be viewed as a circuit-on-demand and is in contrast to the concurrent transmission of multiple flows that compete for available bandwidth which is the philosophy of today’s Internet.

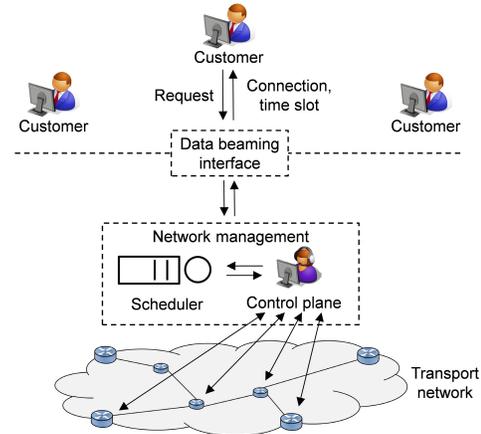


Fig. 1. Overview of the data beaming architecture.

B. Architecture

The data beaming architecture, which is shown in Figure 1, offers interested customers an interface to issue requests for a data beaming slot. In a data beam request, customers indicate the amount of data volume to be transferred, the source and destination site, certain bandwidth requirements, and an optional time window in which the data beam should be executed. Incoming data beam requests are passed to the network management, which consists of a scheduler and the control plane. Both entities work together to determine a transmission path through the network and a free time slot for each data beam. The returned path information may be the label of an existing MPLS label switched path or an existing GMPLS-managed lightpath through a GMPLS/MPLS transport network. The returned data beaming slot and the returned path guarantee that each data beam can be transmitted at the desired rate.

C. Subscription Model

Data beaming guarantees short transmission times at the expense of initial waiting times for a transmission slot. These initial waiting times can be significant if many customers want to transmit multiple data beams. To guarantee an upper bound to the waiting time, the offered load needs to be limited. To that end, we propose a subscription model. A customer may subscribe to the transport network provider and declare how many data beam requests will be issued over time, their bandwidth demands, their directions, and the tolerable waiting time for the transmission slot. Requests for

data beams conforming with these subscriptions are served with high priority, i.e., they face only short waiting times until their data beams can be transmitted. Other customers without subscriptions may also indicate on-demand data beams. On-demand data beams and out-of-profile data beams are served only if no other data beams from in-profile requests are waiting for transmission.

Subscriptions are useful for customers and for providers and may be part of service level agreements. For customers they guarantee a maximum waiting time as long as the requests are in-profile. For providers they serve for planning so that their networks can be provisioned with sufficient resources. To define an upper limit of customers, providers have to consider the distinct paths between possible source and destination sites. The available bandwidth on these paths in combination with the maximum waiting time per customer is then necessary to determine the number of supportable customers. More detailed subscriptions allow more cost-efficient network planning but give less flexibility to the customers. However, those including a fixed source and destination site for example, may be cheaper than more flexible subscriptions, e.g., those specifying only a source site. This is a motivation for customers to make a more detailed subscription.

When requests for data beams conform with existing subscriptions, they must be served within the desired waiting time. To avoid extensive load, the feasibility of new subscriptions or the extension of existing subscriptions must be controlled and they must be explicitly admitted.

D. Simple Scheduler for Data Beaming

The scheduler for data beaming needs to respect the subscriptions of the customers. For simplicity reasons, the data volume offered by each customer is only limited by an average transmission rate r_i , $0 \leq i < n$. That means, if customers send traffic not faster than declared, their data beam requests should be served within short time. In contrast, if customers send bursts of requests or requests for large data beams, they must accept longer waiting times until they can transmit traffic.

For this evaluation, we simplified the considered scheduler to focus on its ability to ensure the subscriptions of the customers. We chose a priority queue that stores data beam requests for a single transmission resource until the transmission starts. In addition, the following algorithm determines the priority dates of requests x . Similar to weighted fair queuing, the algorithm uses virtual finish times $f(x)$ for that purpose, but it is also inspired by VirtualClock's ability to respect reserved rates [8]. For each customer i , the finish time of its last request is recorded in f_i which is initialized with $f_i = -\infty$. Each request is associated with a data volume $b(x)$. When a new request x arrives from customer i , the scheduler calculates the virtual finish time by

$$f(x) = \max(f_i, \text{now}) + \frac{b(x)}{r_i}, \quad (1)$$

whereby *now* is the current time. Then, the customer-specific virtual finish time is set to $f_i = f(x)$ and the request is

inserted into the priority queue using $f(x)$ as priority date. Whenever the transmission resource is free, the request with the lowest priority date is removed from the priority queue and transmitted.

IV. PERFORMANCE MODELING

We first describe a general application scenario and then we develop a performance model for data beaming (DB) and concurrent transmission (CT).

A. Application Scenario

Figure 2(a) shows a transport network interconnecting n customers at different sites. The customer of each site wants to transmit traffic to the other sites which requires a common resource. The common resource may be the transport network of a service provider or adjacent networks which federate to offer data beaming. For this first simulation study, we concentrate on the first case and assume a single provider network as connecting resource between customer sites (see Figure 2(a)). We further treat the service provider network as a single resource as the objective is to present first performance trade-offs rather than a detailed protocol simulation in a realistic topology.

The inter-arrival time between consecutive transmission requests of customer i , $0 \leq i < n$, is modeled by identical and independently distributed (i.i.d.) random variables A_i , and the request sizes are modeled by i.i.d. random variables B_i . With $\lambda_i = 1/E[A_i]$, the overall inter-arrival time can be calculated by $\lambda = \sum_{0 \leq i < n} \lambda_i$.

To simplify our performance evaluation, we assume that all customers have the same mean request inter-arrival time $E[A]$ and mean request size $E[B]$ so that we can calculate the resource utilization by

$$\rho = \frac{\lambda \cdot E[B]}{C}, \quad (2)$$

where C is the transmission capacity of the considered resource. This is visualized in Figure 2(b).

The capacity C and the average request size $E[B]$ give a lower bound on the mean transmission time which equals the mean transmission time $E[T_{DB}] = \frac{E[B]}{C}$ for data beaming. To make our study independent of assumptions of C and $E[B]$, we normalize all performance metrics in our study by $E[T_{DB}]$. The normalization allows the reader to adapt the quantities to any transmission size. In our experiments we choose a number of customers n and a resource utilization ρ , and adjust the mean inter-arrival time $E[A]$ accordingly.

B. Data Beaming: G/G/1-PRIO Model

We assume n customers of a data beaming service with identical subscriptions. The scheduling mechanism for data beaming is configured with $r_i = E[B]/E[A]$ which is the mean traffic rate of the flow and which is also the declared traffic rate in the subscription of the customer. When a customer sends a request to the data beaming interface, the scheduler calculates a priority date for the request according to Equation (1), and the request is inserted into the priority queue according to this

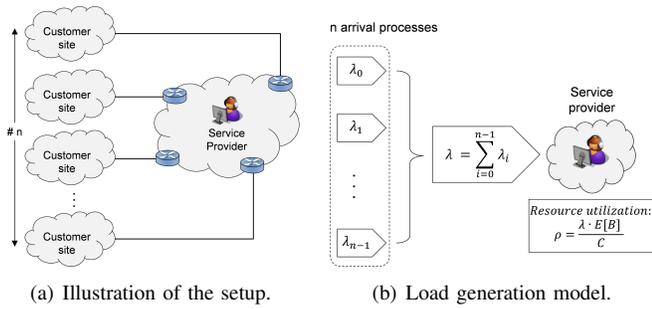


Fig. 2. Application scenario.

priority date, which is then served in a FIFO manner. The service time for data beaming is given by $T_{DB} = B/C$ and the completion time $V_{DB} = W_{DB} + T_{DB}$ of a request is its waiting time W_{DB} plus the service time (see Figure 3(a)).

In the special case that the inter-arrival times A are exponentially distributed, the mean of the waiting time for an M/G/1-FIFO queue can be calculated with the Pollaczek-Khinchin mean value formula (Formula 1.82 in [9]) by

$$E[W] = E[T_{DB}] \cdot \frac{\rho}{1-\rho} \cdot \frac{1 + (c_{var}[B])^2}{2}. \quad (3)$$

This, however, is just an upper bound on the waiting time for data beaming because the scheduler effects that short requests are served earlier than long request which decreases the mean waiting time $E[W_{DB}]$ for data beaming.

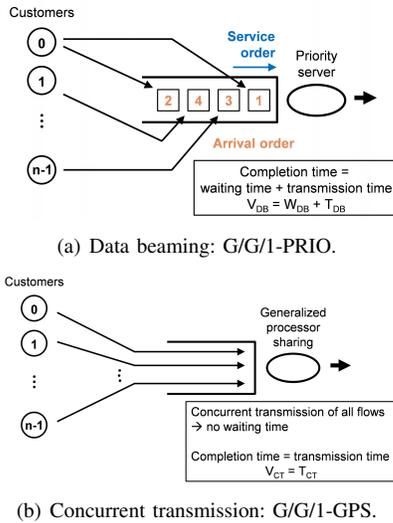


Fig. 3. Queuing models for concurrent transmission and data beaming.

C. Concurrent Transmission: G/G/1-GPS Model

With concurrent transmission, flows are immediately served and fairly share the common resource. This is an optimistic approximation of bandwidth sharing using congestion control algorithms like TCP and stands for the current Internet. Each flow is considered to adjust its sending rate so that the shared network resource is not congested. Bandwidth sharing using TCP relies on the fairness of each flow and does not need a control entity like data beaming.

We model the bandwidth sharing by the generalized processor sharing (GPS) [10] queuing discipline (see Figure 3(b)). It approximates the long-time average of TCP's bandwidth sharing for long-lived flows with equal round trip times. Requests are concurrently served and get an equal amount of the available bandwidth. The shared bandwidth extends the transmission time T_{CT} for concurrent transmission to values that are a multiple of the short transmission time T_{DB} for data beaming. With concurrent transmission, the completion time V_{CT} of a request equals its transmission time T_{CT} .

If the inter-arrival times A are exponentially distributed, the transmission time T_{CT} can be analytically computed according to Formula 4.17 in [9] as

$$E[T_{CT}] = \frac{E[T_{DB}]}{1-\rho}. \quad (4)$$

V. PERFORMANCE COMPARISON

In this section, we analyze the performance of data beaming and concurrent transmission for a single transmission resource. We explain the simulation setup and introduce the completion time V as performance metric. First, we perform experiments where all customers have the same model for transmission requests and then we study how an in-profile and an out-of-profile class of customers compete for the transmission resource.

A. Simulation Setup

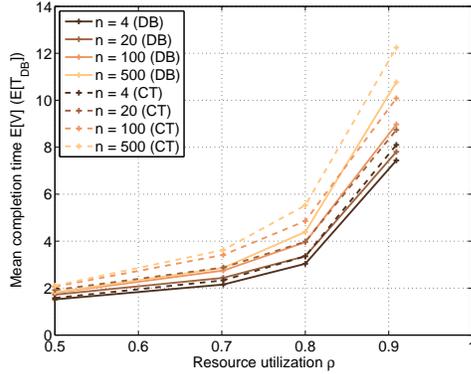
We simulate data beaming and concurrent transmission using a flow-based model in OMNeT++. In both cases, we simulate transmission requests from n different customers. To model the inter-arrival time A of consecutive transmission requests of a customer and the request size distribution B , we use a Gamma distribution as we can easily control its mean and coefficient of variation. For $c_{var} = 1.0$, the Gamma distribution becomes an exponential distribution which we use as default value in some cases. We avoid coefficients of variations of inter-arrival times smaller than $c_{var}[A] = 0.1$ as those systems tend to become quasi-periodic so that they require extremely long simulation runs to provide reliable results. This is not problematic for request sizes and we use a deterministic distribution for $c_{var}[B] = 0$, i.e., all requests have the same size. We simulate all experiments so long that 95% confidence intervals are smaller than 1% of the simulated value, but for the sake of clarity, we omit them in the presented figures.

B. Performance Metric

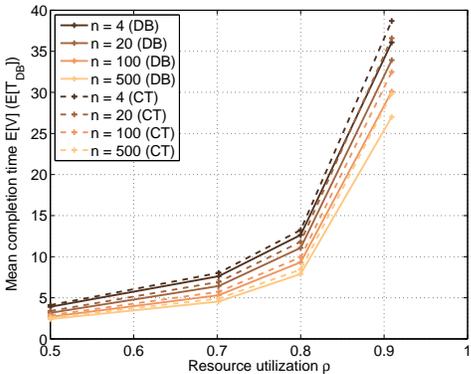
In the following, we consider the mean completion time $E[V]$ of transmission requests. In case of data beaming it is the sum $E[V_{DB}] = E[W_{DB}] + E[T_{DB}]$ of the mean waiting time and the mean transmission time. For concurrent transmission, it is just the mean transmission time $E[V_{CT}] = E[T_{CT}]$. We present the completion time as a multiple of the minimum mean transmission time which is equal to $E[T_{DB}] = E[B]/C$. Thus, the normalized mean waiting time $E[W_{DB}]$ for data beaming is the normalized mean completion time $E[V_{DB}]$ minus 1 and is hence implicitly given in each figure.

C. Customers with Equal Transmission Request Models

We assume that n customers send traffic over the single resource and that each of them has a subscription for a traffic rate of $\frac{E[B]}{E[A]}$. The obvious system parameters are the number of customers n and the resource utilization ρ . In addition, the coefficient of variation of the inter-arrival time $c_{var}[A]$ and the transmission request size $c_{var}[B]$ influence the completion times for data beaming and concurrent transmission. We investigate their impact in the following.



(a) The coefficient of variation for inter-arrival times of transmission requests is $c_{var}[A] = 0.1$.



(b) The coefficient of variation for inter-arrival times of transmission requests is $c_{var}[A] = 2.0$.

Fig. 4. The completion time for data beaming and concurrent transmission depends mainly on the resource utilization ρ but only little on the number of customers n . The coefficient of variation of the transmission request sizes is $c_{var}[B] = 1.0$.

1) *Impact of Resource Utilization, Number of Customers, and Inter-Arrival Time Variability:* In our first experiments, we set the coefficient of variation of the transmission request size to $c_{var}[B] = 1$ and set the one for the inter-arrival time to extreme values $c_{var}[A] = 0.1$ and $c_{var} = 2.0$. Figures 4(a) and 4(b) show the completion time depending on the resource utilization ρ and the number of customers n . In both cases, the completion time increases with increasing resource utilization and it is shorter for data beaming than for concurrent transmission. The transmission time for concurrent transmission $E[V_{CT}]$ is at least twice as large as the one for data beaming $E[T_{DB}]$ and can quickly become a multiple of it. The figures show that for $c_{var}[A] = 0.1$, the completion time slightly increases and for $c_{var}[A] = 2.0$ slightly decreases with the number of customers n , but the impact of the utilization ρ is much larger.

2) *Impact of Request Size Variability:* In another experiment we observed that the completion time is independent of the number of customers n for $c_{var}[A] = 1.0$. In that case, the inter-arrival times between requests of a single customer are exponentially distributed as we use a Gamma distribution. As a consequence, the inter-arrival times from all customers are also exponentially distributed. As we keep the resource utilization ρ constant, the mean of the inter-arrival time of the request arrivals multiplexed from all customers is the same for all n . As the exponential distribution has only a single parameter, the arrival process for all requests is the same for all n . Therefore, the simulation results are independent of n for $c_{var}[A] = 1.0$.

Figure 5 shows the completion times for data beaming and concurrent transmission depending on the resource utilization ρ for different coefficients of variations $c_{var}[B] \in \{0.1, 1.0, 2.0\}$ of the transmission request sizes. We observe that the completion time for concurrent transmission is independent of that value. In fact, for the special case of exponentially distributed inter-arrival times, the completion time can also be analytically calculated by Equation (4) which is also independent of the coefficient of variation $c_{var}[B]$. We also observe that the completion time for data beaming increases with increasing $c_{var}[B]$ and it even exceeds the one for concurrent transmission for a large value of $c_{var}[B] = 2.0$.

For $c_{var}[A] = 1.0$ and $c_{var}[B] = 1.0$, concurrent transmission leads to the same mean waiting time as an M/G/1 queuing system. In Figure 5, the curve for the completion time with data beaming is lower than the one with concurrent transmission. This shows that the completion time for M/M/1 systems can be reduced by substituting the FIFO service order by the order proposed by our scheduler.

3) *Overview of Impact of Transmission Request Variability:* We systematically study the effect caused by the coefficients of variations $c_{var}[A]$ and $c_{var}[B]$ for $n = 100$ and $\rho = 0.7$. Figure 6 shows the completion time depending on the coefficient of variation of the inter-arrival time $c_{var}[A]$ for different coefficients of variation $c_{var}[B]$ of the transmission request size. Increasing variability of the inter-arrival time leads to longer

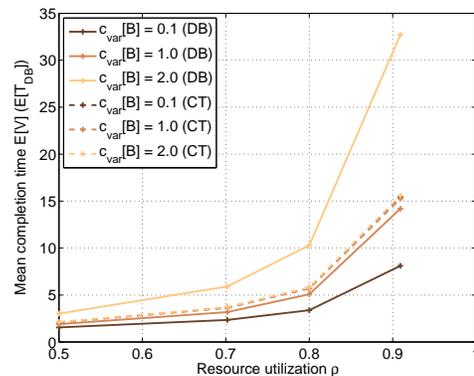


Fig. 5. The completion time for exponentially distributed inter-arrival times of transmission requests is independent of the number of customers n . It depends only for data beaming on the variability of the transmission request sizes.

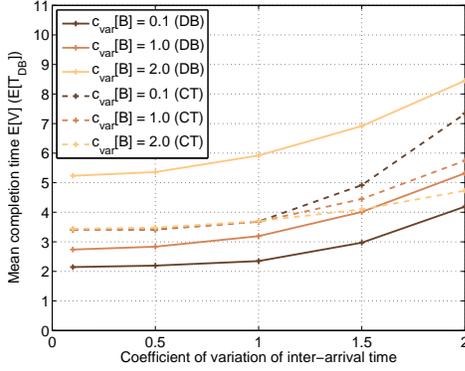


Fig. 6. The completion time for data beaming and concurrent transmission depends on the coefficient of variations $c_{var}[A]$ and $c_{var}[B]$ for the request inter-arrival time A by a customer and the request size B ($\rho = 0.7, n = 100$).

completion times. For data beaming, more variable transmission request sizes also increase the completion time. But for concurrent transmission we observe that the completion time is identical as long as the coefficient of the inter-arrival time is at most $c_{var}[A] = 1.0$. If the coefficient of variation $c_{var}[A]$ is larger, the completion time decreases with an increasing coefficient of variation $c_{var}[B]$ of the transmission request size. This sounds counterintuitive, but can be explained as follows. If a transmission request is very long, many very short requests may arrive in the meantime, they are served in parallel, they complete quite quickly as they are short, and in particular earlier than the long transmission request. The average completion time may be short if many short transmission requests arrive. With increasing variability of the transmission request size, the effect of this example dominates the completion time.

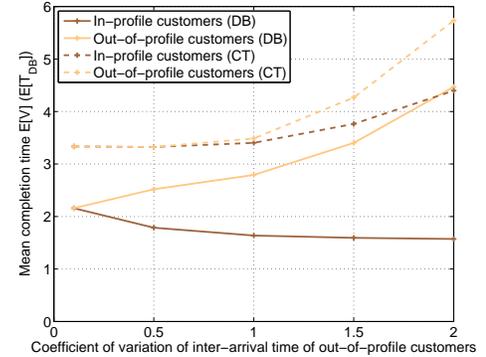
D. Customers with Equal Transmission Request Models

In the following, we call customers in-profile if they send transmission requests smoothly over time so that their transmission rates deviate only little from their subscribed rate. Otherwise, we call them out-of-profile. We only need this qualitative description in the following to differentiate two customer types.

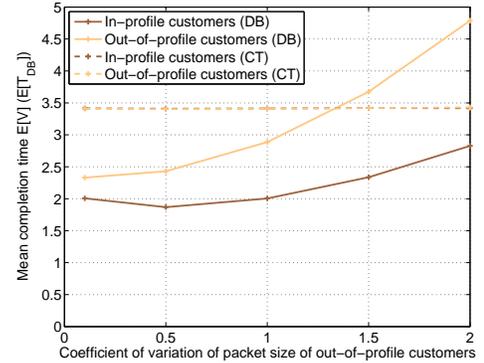
We have shown that data beaming leads to short waiting times when all customers are in-profile, i.e. when they have low $c_{var}[A]$ and low $c_{var}[B]$, and that data beaming leads to long waiting time when all customers are out-of-profile. Now we assume that a class of in-profile customers ($c_{var}[A] = 0.1, c_{var}[B] = 0$) and a class of out-of-profile customers compete for the transmission resources. We show that our proposed scheduler for data beaming can well enforce short waiting times for in-profile customers under various conditions. We use $n = 100$ customers in the following experiments and a resource utilization of $\rho = 70\%$.

1) *Coexistence of In-Profile Customers and Different Out-of-Profile Customers:* In our next experiments, we have 50% in-profile customers and 50% out-of-profile customers. We first consider out-of-profile customers that have constant request sizes ($c_{var}[B] = 0$) and vary the coefficient of variation of their inter-arrival times $c_{var}[A]$. Figure 7(a) shows that the

completion time for in-profile customers decreases with the variability of the traffic sent by out-of-profile customers and that the completion time for out-of-profile customers increases. This is due to the fact that in-profile customers have higher priority over more out-of-profile customers so that they get served faster when the transmission resource becomes free again. An important observation is that the completion time of in-profile customers is bound by 2.2 times the mean transmission time $E[T_{DB}]$ for data beaming. For concurrent transmission, the completion time is much larger and transmission requests from in-profile customers face similar waiting times as transmission requests from out-of-profile customers.



(a) $c_{var}[A]$ is variable, $c_{var}[B] = 0$.



(b) $c_{var}[A] = 0.1, c_{var}[B]$ is variable.

Fig. 7. Completion time for data beaming and concurrent transmission for 50% in-profile and 50% out-of-profile customers with different characteristics ($\rho = 0.7, n = 100$).

Now, we study out-of-profile customers that have low variability in the inter-arrival times of consecutive requests ($c_{var}[A] = 0.1$) and vary the coefficient of variation of their request sizes $c_{var}[B]$. Figure 7(b) shows that with data beaming the completion time for both in-profile and out-of-profile customers increases with the variability of the size of the requests sent by out-of-profile customers. However, out-of-profile customers need to wait much longer and the completion time of requests from in-profile customers is bounded by 2.8 times the mean transmission time $E[T_{DB}]$. The reason for the increase of the completion time for all customers is the fact that the transmission resource takes longer to be released when long requests are served which is more likely with more variable request sizes. With concurrent transmission, the completion time is independent of the variability of the

request sizes and it is the same for in-profile and out-of-profile customers. In particular, it is significantly larger than the completion time for in-profile customers with data beaming.

2) *Varying Percentage of Out-of-Profile Customers*: In the next experiment, the out-of-profile customers have a coefficient of variation of the transmission request inter-arrival time of $c_{var}[A] = 2$ and of the request size of $c_{var}[B] = 2$. Figure 8 shows the completion time for in-profile and out-of-profile customers for different percentages of out-of-profile customers. For in-profile data beaming customers, the completion time is not larger than 2.9 times the mean transmission time $E[T_{DB}]$ while it is almost three times larger for out-of-profile customers than for in-profile customers. This is different with concurrent transmission where the completion time for in-profile and out-of-profile customers does not differ a lot.

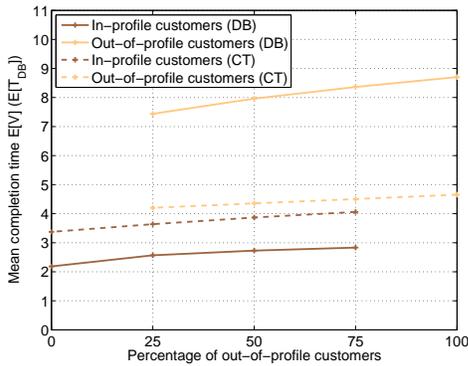


Fig. 8. Data beaming leads to short completion times for in-profile customers and to long completion times for out-of-profile customers while with concurrent transmission the completion time for both customer types does not differ a lot ($\rho = 0.7, n = 100$).

3) *Summary*: We have shown that our proposed scheduler for data beaming organizes the transmission of data beams in such a way that customers face only short waiting times if they are in-profile with regard to their subscriptions. If a customer wants to send more bursty traffic, it needs larger subscriptions to keep its waiting times low. Customers do not even need to have subscriptions, but then they should be served only if there is no other request from a subscribed customer in the queue. As a result, such customers would face even longer completion times.

Thus, data beaming subscriptions are useful for customers if they want to have short waiting times until transmission. This is a kind of priority service and may be charged. Moreover, they give hints for resource provisioning to the network operator.

VI. RELATED WORK

In this section, we present work related to data beaming. We first describe the similar concept of advance reservations and then introduce other architectures that also support reservation of network resources in advance and explain, how data beaming differs from them.

A. Advance Reservation

Requests for an advance reservation of network resources usually contain a source and destination node, a certain

bandwidth requirement, and a specific time window during which the reservation becomes active. In contrast, requests for immediate reservation do not contain a time window and become immediately active once they have been admitted. With respect to the exact specification of the time window, the authors of [11] present a taxonomy to characterize advance reservation requests into three different categories.

Specific start and duration (STSD): customers indicated a specific starting time and a specific duration and do not tolerate any time displacement.

Specific start and unspecified duration (STUD): customers only specify the starting time and expect to get the desired network resource as long as possible.

Unspecified start and specific duration (UTSD): customers specify the duration of an advance reservation but do not specify a starting time. The customers in this case expect to get the desired network resource as soon as possible.

Data beaming can be categorized as a special case of UTSD where a customer indicates a maximum ending time for the advance reservation. The customers tolerate a certain delay until the communication starts as long as the data transmission is finished before the specified maximum ending time.

The flexibility of data beaming with respect to the time window avoids a big issue of advance reservations. Strict advance reservation reduces the resource utilization and acceptance rate due to a fragmentation of available network resources [12]. Data beaming loosens the strict time window constraints of advance reservation requests and hence improves the resource utilization and acceptance rate. The correlation between the laxity of advance reservations and an improved resource utilization and acceptance rate has been shown for example in [13] and [14].

B. Architectures with Similar Ideas

The GridFTP protocol [15] is part of the Globus Toolkit (GT) [16] and used for data access and movement. It is an extension of the FTP protocol and offers a reliable and high performance data transfer. In comparison to data beaming, GT is rather a protocol suite used to build grid networks than a new network service concept like data beaming. Both concepts enable the transfer of large data volumes, whereas GT is intended to enable applications with federated resources and data beaming primarily offers a new network service for customers which require fast and reliable data transfer at a very high bandwidth.

The framework proposed in [7] extends the GMPLS suite and its routing and signaling protocols to support advance reservation of network resources. The objective is to enable automated provisioning of advance reservations in a GMPLS-based network. The framework covers only the network reservation aspect of the data beaming architecture and does not propose a full architecture with network management. However, the framework could be used as underlying signaling protocol to realize the data beaming architecture on top of a GMPLS-based transport network.

The concept of a bandwidth broker as network management entity for a GMPLS-based network is used in the DRAGON architecture [17]. The objective is to provide large data rates for file transfer in optical networks. However, in contrast to data beaming, resources are reserved for immediate use and not in advance. The architecture specified in [18] also uses a bandwidth broker to support flexible advance reservations based on a MPLS network. The reservation requests allow certain bandwidth constraints that specify a minimum and a maximum bandwidth and a total deadline for the data transmission. This concept is very similar to data beaming.

However, the main difference between these architectures and our data beaming approach is that they do not consider a subscription model for resource reservation requests. The subscription model in data beaming facilitates the resource provisioning from the provider's perspective. Requests for a reservation of a data beaming slot are only considered if they conform to the current subscription of the customer. Otherwise, the data beaming scheduler does not guarantee that a data beam request can be realized. This is a significant improvement and both interesting for providers and customers.

VII. CONCLUSION

In this work, we explained the concept of data beaming together with a subscription model and proposed a simple scheduler. Customers indicate transmission requests and receive time slots for high-speed data transfers after some time. It is useful for applications that require high data rates but can wait some time until transmission starts.

In the current Internet, flows are transmitted concurrently and receive only a fraction of the physical bandwidth. Our simulations showed that the completion time of transmission requests is in most cases shorter for data beaming than for concurrent transmission. However, the main benefit of data beaming is that its transmission time takes only a fraction of the one of concurrent transmission.

The subscription model for data beaming helps to provision the network with sufficient resources. A scheduler enforces that transmission requests from customers who conform with their subscriptions are earlier served than transmission requests from customers who exceed their subscriptions or who have no subscriptions at all. Our simulations showed that the waiting time for in-profile customers is low, also in the presence of out-of-profile customers. The completion time for in-profile users is also clearly shorter with data beaming than with concurrent transmission. Thus, subscriptions for data beaming offer a priority service to customers and may be charged.

Open issues are the extension to a more realistic scenario as well as considering alternative scheduling algorithms, e.g. from the related work area. Furthermore, a detailed protocol specification for data beam reservation, like in [19], is currently out of the scope and will be addressed in a future work. Data beaming in this context may enable the network provider to offer new high-speed services to customers based on its flexible network infrastructure like GMPLS without giving the customer control over its network.

REFERENCES

- [1] F. Checconi, T. Cucinotta, and M. Stein, "Real-Time Issues in Live Migration of Virtual Machines," 4th Workshop on Virtualization in High Performance Computing (VHPC'09), Delft, The Netherlands, Sep. 2009.
- [2] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *USENIX Symposium on Networked Systems Design & Implementation (NSDI)*. Berkeley, CA, USA: USENIX Association, 2008, pp. 337–350.
- [3] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schiöberg, "Live wide-area migration of virtual machines including local persistent state," in *International Conference on Virtual Execution Environments*. New York, NY, USA: ACM, 2007, pp. 169–179.
- [4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *USENIX Symposium on Networked Systems Design & Implementation (NSDI)*. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286.
- [5] A. Stage and T. Setzer, "Network-aware migration control and scheduling of differentiated virtual machine workloads," in *ICSE Workshop on Software Engineering Challenges of Cloud Computing*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 9–14.
- [6] W. Imajuku, E. Oki, R. Papneja, S. Morishita, K. Ogaki, M. Miyazawa, K. Miyazaki, H. Nakazato, H. Sugiyama, J. Allen, S. Hasegawa, N. Sakuraba, I. Nishioka, S. Seno, Y. Nakahira, D. Ishii, S. Okamoto, T. Unen, M. Blumhardt, H. Rakotoranto, and V. Pandian, "A multi-area MPLS/GMPLS interoperability trial over ROADM/OXC network," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 168–175, Feb. 2009.
- [7] J. Zheng, B. Zhang, and H. Mouftah, "Toward automated provisioning of advance reservation service in next-generation optical internet," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 68–74, Dec. 2006.
- [8] L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks," *ACM SIGCOMM Computer Communications Review*, vol. 20, no. 4, pp. 19–29, 1990.
- [9] L. Kleinrock, *Queueing Systems*, 1st ed. New York: John Wiley and Sons, 1976, vol. 2: Computer Applications.
- [10] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single-Node Case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, 1993.
- [11] J. Zheng and H. T. Mouftah, "Routing and Wavelength Assignment for Advance Reservation in Wavelength-Routed WDM Optical Networks," in *IEEE International Conference on Communications (ICC)*, vol. 5, Aug. 2002, pp. 2722–2726.
- [12] L.-O. Burchard, H.-U. Heiss, and C. A. F. De Rose, "Performance Issues of Bandwidth Reservations for Grid Computing," in *Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. Washington, DC, USA: IEEE Computer Society, 2003, p. 82.
- [13] U. Farooq, S. Majumdar, and E. Parsons, "Dynamic scheduling of light-paths in lambda grids," in *IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS)*, Oct. 2005, pp. 1463–1472 Vol. 2.
- [14] E. He, X. Wang, and J. Leighton, "A Flexible Advance Reservation Model for Multi-Domain WDM Optical Networks," in *IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS)*, San Jose, CA, 2006, pp. 1–10.
- [15] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, "Data Management and Transfer in High-Performance Computational Grid Environments," *Parallel Computation Journal*, vol. 28, pp. 749–771, 2001.
- [16] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," in *International Conference on Network and Parallel Computing (IFIP)*, vol. 21, no. 4, Jul. 2006, pp. 513–520.
- [17] T. Lehman, J. Sobieski, and B. Jabbari, "DRAGON: a framework for service provisioning in heterogeneous grid networks," *IEEE Communications Magazine*, vol. 44, no. 3, pp. 84–90, Mar. 2006.
- [18] L.-O. Burchard, "Networks with Advance Reservations: Applications, Architecture, and Performance," *Journal of Network and Systems Management*, vol. 13, pp. 429–449, 2005.
- [19] D. Kuo and M. Mckeown, "Advance reservation and co-allocation protocol for grid computing," in *First International Conference on e-Science and Grid Computing*, Jul. 2005, pp. 164–171.