PACCP: A Price-Aware Congestion Control Protocol for Datacenters

Xiaocui Sun
Guangdong Pharmaceutical University
Guangzhou, China
Email: xiaocuisun1002@hotmail.com

Zhijun Wang
The University of Texas at Arlington
Arlington, USA
Email: zhijun.wang@uta.edu

Yunxiang Wu
Purple Mountain Laboratories
Nanjing, China
Email:wuyunxiang@pmlabs.com.cn

Hao Che
The University of Texas at Arlington
Arlington, USA
Email: hche@cse.uta.edu

Hong Jiang
The University of Texas at Arlington
Arlington, USA
Email: hong.jiang@uta.edu

***Abstract*—To date, customers using infrastructure-as-a service (IaaS) cloud services are charged for the usage of computing/storage resources, but not the network resource. The difficulty lies in the fact that it is nontrivial to allocate network resource to individual customers effectively, especially for short-lived flows, in terms of both performance and cost. To tackle this challenge, in this paper, we propose PACCP, an end-to-end Price-Aware Congestion Control Protocol for cloud services. PACCP is a network utility maximization (NUM) based optimal congestion control protocol. It supports three different classes of services (CoSes), i.e., best effort service (BE), differentiated service (DS), and minimum rate guaranteed (MRG) service. In PACCP, the desired CoS or rate allocation for a given flow is enabled by properly setting a pair of control parameters, i.e., a minimum guaranteed rate and a utility weight, which in turn, determines the price paid by the user of the flow. Two pricing models, i.e., a coarse-grained Virtual machine (VM)-Based Pricing model (VBP) and a fine-grained Flow-Based Pricing model (FBP), are proposed. PACCP is evaluated by both large scale simulation and small testbed implementation. The results demonstrate that PACCP provides minimum rate guarantee, high bandwidth utilization and fair rate allocation, commensurate with the pricing models.**

***Keywords*-Price model, cloud computing, congestion control**

I. Introduction

An infrastructure-as-a-service (IaaS) cloud, such as Amazon EC2 and Alibaba cloud, provides scalable, pay-as-you-go computing resources to its customers. However, to date, the customers using such cloud services are charged based on the usage of the computing related resources only, e.g., various instances of virtual machines (VM) with different CPU. This, however, is inadequate, as paying for a given VM instance provides no assurance of flow performance for a flow emitted from that instance [1]. The root cause of the status quo is that the network bandwidth is shared in a highly dynamic environment by flows emitted from all VM instances and hence, it is difficult to provide quantifiable flow rate allocation in a cost-effective fashion so that an effective pricing structure can be built around it. A direct consequence for not being able to do so is that a customer may experience poor performance, especially at high network utilization, incommensurate with the price the customer has paid for the use of the computing resources[1].

To tackle the above challenges, network pricing solutions based on explicit bandwidth reservation have been proposed [2], [3], [4], [5], [6], [7], [8], [9]. The price is usually dynamically generated either through an auction process [3], [5], [7], [8] or a time-varying price table [2], [4], [6], [9], adjusted based on the current and/or historical statistics. However, these pricing solutions are only effective for long-lived flows, such as video on demand [10], not for the popular user-facing datacenter applications [11], [12], [13], [14] which usually have small flow sizes and short durations.

User-facing datacenter applications, such as Web searching [12] and social networking [14], are usually associated with a stringent tail-latency service level objective (SLO) [15]. Moreover, a job for such an application generally involves one or multiple stages of parallel task processing by (many) instances, which generate bursts of (massive) numbers of flows emitted from those instances. Such flows are usually short-lived with sizes of less than 1 Mbytes [16], [17] and with tight flow completion time budget, e.g., a few milliseconds, to meet a prescribed tail-latency SLO.

The pricing solutions mentioned earlier are based on centralized bandwidth reservation, which is either pre-configured or flow-driven, none of which however, can deal with the above workload effectively. On one hand, pre-configured bandwidth reservation that allocates bandwidths for the prospective flows in advance is not scalable and cannot handle bursts of massive numbers of short-lived flows. Moreover, without knowing the flow start time and flow size, this approach may lead to either over or under resource provisioning, causing violation of SLOs or low resource utilization, respectively. On the other hand, flow-driven bandwidth reservation that reserves bandwidth upon a flow arrival is generally too slow due to centralized control to meet the tight flow response time budget of such flows and incurs excessive processing and communication overheads.

978-0-7381-2394-3/20/$31.00 ©2020 IEEE
DOI 10.1109/UCC48980.2020.00022

Moreover, these solutions need significant core network switches modification/upgrading incurring high costs. Although a price-aware distributed scheduling solution, known as SoftBW [1], is proposed to allow scalable bandwidth reservation with flow rate guarantee, it only works at very low network loads (less than 30%), as it reserves bandwidths for individual VM instances at each host port only assuming that the datacenter network is congestion free. Any retransmission can result in flow rate allocation inappropriate for the paid prices.

To overcome the above shortcomings of the existing network pricing solutions, in this paper, we propose a Price-Aware Congestion Control Protocol (PACCP) for IaaS cloud services. PACCP is a network utility maximization (NUM) based optimal congestion control protocol. It supports three different class of of services (CoSes), i.e., best effort service (BE), differentiated service (DS) and minimum rate guaranteed (MRG) service. The three types of services are enabled by properly setting the values of a pair of parameters, i.e., a minimum guaranteed rate and a utility weight, which are, in turn, determined by the flow price paid for the services.

In this paper, we propose two pricing models, i.e., a coarse-grained VM-Based Pricing model (VBP) and a fine-grained Flow-Based Pricing model (FBP). A customer pays a price to buy a desired service, which is then mapped to given values of the pair of parameters in PACCP. PACCP possesses the following salient features,

- It is an optimal solution in terms of NUM; It uses the TCP utility function and hence it is a TCP friendly protocol.
- It meets the three widely accepted requirements for datacenter price-based rate allocation solutions [2], i.e., providing minimum rate guarantee; achieving high network utilization; and allocating flow rates in proportion to the paid prices;
- To the best of our knowledge, it is the first solution that seamlessly integrates pricing models with end-to-end congestion control protocols. Hence, it is high scalable and can deal with bursts of unlimited numbers of short-lived flows. It allows flows to fully utilize all available bandwidths and thus improving the bandwidth utilization. Moreover, it allows adjustment of pricing at runtime, adapting to resource demand changes and/or network load changes;
- It only requires software upgrade in end hosts and does not need any change in core network switches and hence, is readily deployable in today's datacenters.

PACCP is evaluated by large scale simulations as well as a small testbed implementation. The results demonstrate that PACCP can indeed provide soft minimum rate guarantee, high network utilization and rate allocation proportional to the prices paid, hence, meeting all three requirements for datacenter network pricing solutions.

The remainder of the paper is organized as follows. Section II presents the NUM-based flow rate allocation framework. Section III gives the NUM-based optimal congestion control laws. Two pricing models are given in Section IV and evaluated in Section V. Section VI presents the related work. Finally Section VII concludes the paper.

II. Network utility maximization based rate allocation

Assume that a network has n active flows and $U_i(x_i)$ is the user utility function of flow rate x_i for flow i (i=1,2,...,n). Then NUM is defined as the following,

$$V = max\{\sum_{i=1}^{n} U_i(x_i)\}, \tag{1}$$

subject to link bandwidth and flow rate constraints. The goal of NUM is to find distributed flow rate control laws that lead to flow rate allocation, $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$, where the global design objective V is attained, or collective user satisfaction of the services is maximized, as user utilities are meant to characterize to what degree users are satisfied with the services they receive. The traditional NUM solution usually can only work on a single user utility function. Our recently developed NUM solution, called HOLENT NUM [18], can deal with different user utility functions. Clearly, the relative user utilities of the flows determine the rate allocation $\mathbf{x}$, provided that the flow rate constraints are satisfied. In other words, in NUM, the fairness criterion is uniquely determined by the relative user utilities of the flows. While minimum flow rate requirements can be easily enforced as flow rate constraints in NUM, it is nontrivial to enable flexible and quantifiable fairness criteria. In what follows, we propose a solution based on weighted user utilities.

We consider the following weighted utility function, i.e., $U_i(x_i) = w_i U_0(x_i)$, where U_0 is a base utility function shared by all the flows and w_i is the weight of flow i (i=1,2,...,n). To be backward compatible with and friendly to TCP flows, we use the TCP utility function (U_{TCP}), which is concave, as the base utility function. The utility function for TCP Reno is derived in [19], [20] and is given as follows. In the slow start phase (SSP),

$$U_{TCP}(x) = xlog(1 + \frac{\alpha}{\beta}), \tag{2}$$

and, in congestion avoidance phase (CAP),

$$U_{TCP}(x) = (\frac{\mu}{\beta} + x)[log(\mu + \beta x) - 1] - x[log(\beta x) - 1], \tag{3}$$

where αx and βx are the multiplicative increase and decrease rates, respectively. To match with SSP in TCP Reno where the flow rate is doubled/halved every round trip time (RTT), we have α=2β=$1/RTT$, by approximating the increase and decrease rates to be constant within a RTT interval. μ is the additive-increase rate (i.e., the rate of one

packet per RTT) in CAP. With this TCP utility, now NUM can be rewritten as,

$$V = max\{\sum_{i=1}^{n} w_i U_{TCP}(x_i)\}, \tag{4}$$

subject to the link bandwidth and minimum flow rate constraints.

Now the idea is to enable flexible fair flow rate allocation through weight assignment. Specifically, consider flows with different weights sharing a bottleneck link. With the Lagrangian multiplier technique [21], it can be easily shown that the rate allocation that satisfies V meets the following condition[1],

$$\frac{w_i}{w_j} = \frac{dU_{TCP}(x_j)/dx_j}{dU_{TCP}(x_i)/dx_i} = \frac{log(1+\mu/\beta x_j)}{log(1+\mu/\beta x_i)} \approx \frac{x_i}{x_j}, \quad \forall i,j, \tag{5}$$

for any pair of flows i and j bottlenecked at this link. Here we assume $\beta x \gg \mu$, i.e., the multiplicative decrease rate (i.e., half of the flow rate) is much larger than the additive increase rate (rate of 1 packet per RTT), and hence $log(1+\mu/\beta x) \approx \mu/\beta x$. Eq. (5) clearly indicates that the allocated flow rate ratio is proportional to their utility weight ratio for any two flows sharing a bottleneck link. Hence, the relative flow rates allocated to different flows can be easily adapted to the fairness criterion underlaid by any given pricing model, through the proper setting of the corresponding relative weights.

III. NUM BASED OPTIMAL CONGESTION CONTROL LAWS

A family of optimal congestion control laws to NUM with concave user utilities are derived by Su, et. al. [22], which underpins PACCP. Now we first introduce this family of optimal congestion control laws, which is then applied to the NUM problem in Eq. (4) to derive PACCP.

For simplicity, the subscript i for user i is skipped hereafter. For any flow with concave utility function $U(x)$, the family of optimal congestion control laws that satisfies V are,

$$\dot{x} = z(x,t,cg)[f(x) - (1-\bar{cg}r(x))] \tag{6}$$

with

$$f(x) = 1 - e^{-dU(x)/dx}, \tag{7}$$

where $z(x,t,cg)$ can be any piecewise continuous positive scaler function, resulting in an unlimited number of possible control laws in the family; cg is the path congestion indicator, taking value 1, if the path is congested, and 0, otherwise; $\bar{cg}$ is the logical negation of cg; $r(x)$ is a scalar parameter associated with the minimum rate requirement. Assume that a flow has a minimum rate requirement θ, i.e., $x \geq \theta$. Then $r(x)$ is given as,

$$r(x) = \begin{cases} 1 & if \quad x \geq \theta \\ r_{cos} & if \quad x < \theta, \end{cases} \tag{8}$$

with $r_{cos} > 1$, a design parameter. We suggest to use r_{cos}=3 according to our performance studies of this parameter (due to the page limitation, the results are not shown here).

For example, it can be easily shown that by applying the above family of optimal congestion control laws to the TCP utility in Eqs. (2) and (3) and let, $z(.) = z_{TCP}(x,t,cg)$, where,

$$z_{TCP}(x,t,cg) = \begin{cases} (\alpha+\beta)x & for \quad SSP \\ \mu+\beta x & for \quad CAP. \end{cases} \tag{9}$$

we arrive at the TCP Reno congestion control law [19].

Now applying the above family of optimal congestion control laws to the NUM problem given in Eq. (4), we arrive at PACCP as follows. In the SSP,

$$\dot{x} = \begin{cases} (3r(x) - 3^{1-w})\alpha x/2 & if \quad cg = 0 \\ -3^{-w+1}\beta x & if \quad cg = 1, \end{cases} \tag{10}$$

and in the CAP,

$$\dot{x} = \begin{cases} [-(\frac{\beta x}{\mu+\beta x})^w + r(x)](\mu+\beta x) & if \ cg = 0 \\ -(\frac{\beta x}{\mu+\beta x})^w(\mu+\beta x) & if \ cg = 1, \end{cases} \tag{11}$$

Clearly, flow rate allocation is determined by $r(x)$ (or θ) and w, a pair of parameters that uniquely determine PACCP.

To be backward compatible with TCP window-based congestion control, we translate the fluid-flow based control laws in Eqs. (10) and (11) into a window-based congestion control protocol. In the window-based control, the flow rate stays unchanged during each RTT interval. Hence, the congestion window can be calculated as $W_c = xRTT/MSS$, where MSS is the maximum segment size. The flow rate change from one RTT epoch to the next RTT epoch is Δx=$\dot{x}RTT$, where $\dot{x}$ is the flow rate change over an RTT epoch, which can be estimated by the fluid-flow control law. Hence, the window size change ΔW_c at the end of every RTT epoch is calculated as $\Delta W_c = \Delta x RTT/MSS$. The minimum congestion window size W_{min} corresponding to a minimum guaranteed rate θ is given as follows,

$$W_{min} = \frac{\theta RTT}{MSS}. \tag{12}$$

For a flow without minimum rate requirement, $W_{min} = 0$ (i.e., θ=0).

Define $CND1$ as $\{ cg = 0 \ \& \ W_c < W_{min}\}$ and $CND2$ as $\{cg = 0 \ \& \ W_c \geq W_{min}\}$. Now the window-based protocol for congestion window size (W_c) update (i.e., $W_c = W_c + \Delta W_c$) can be approximated (by assuming $\beta x \gg \mu$) as follows. In the SSP,

$$W_c = \begin{cases} ((3r_{cos} - 3^{1-w})/2 + 1)W_c & if \quad CND1 \\ ((3 - 3^{1-w})/2 + 1)W_c & if \quad CND2 \\ (1 - 3^{1-w}/2)W_c & if \quad cg = 1, \end{cases} \tag{13}$$

[1]Here we apply the CAP TCP utility, not the SSP TCP utility, because the TCP timeout is a rare event and TCP runs in the congestion avoidance phase most of the time.

and in the CAP,

$$W_c \approx \begin{cases} \frac{1}{2}(r_{cos}+1)W_c + (r_{cos}-1+w) & if\ CND1 \\ W_c + w & if\ CND2 \\ \frac{1}{2}W_c + (w-1) & if\ \ cg = 1. \end{cases} \quad (14)$$

Note that TCP Reno is a special case of the above control protocol with r_{cos}=1 (i.e., θ=0) and w=1. A flow under the control of the above PACCP receives minimum rate guarantee and quantifiable fair sharing of the additional bandwidth. PACCP supports three broad CoSes based on specific settings of the pair of parameters (θ, w), i.e., the best effort (BE) service with (0, 1) (i.e., TCP Reno); the differentiated service (DS) with (0, $w>1$); and the minimum rate guaranteed (MRG) service with ($\theta>0$, $w \geq 1$). The three CoSes can be enabled by simply passing a pair of control parameters, i.e., (θ, w), into PACCP. Hence, pricing models tied to this pair of control parameters may be developed to charge users in proportion to the (relative) network bandwidths allocated to them. For datacenter with both TCP and UDP traffic, the two parameters can also be enabled to DCCP congestion control protocols [23], [24] to provide the three different types of services for UDT traffic. The only difference is that there is no retransmission in UDP traffic in case of congestions.

IV. Pricing model

In this section, we discuss how to set the pair of parameters based on the flow prices in PACCP to support the three CoSes. Two pricing models, a coarse-grained VM-Based Pricing model (VBP) and a fine-grained Flow-Based Pricing model (FBP), are proposed to suppor the three CoSes.

In VBP, a user paying for the usage of a VM instance will also pay a network usage fee per unit time for an aggregated bandwidth determined by a given pair, (θ, w). In this model, in principle, each VM instance can support more than one CoS, as long as, $\sum_i^{n_v} \theta_i \leq \theta$ and $\sum_i^{n_v} w_i \leq w$ (it can be easily shown that both parameters are additive), where n_v is the number of active flows emitted from the instance and (θ_i, w_i) is the pair of control parameter for flow i, for $i = 1, ..., n_v$. Namely, the only requirement is that the network bandwidth taken by all the flows emitted from this instance is upper bounded by the aggregated bandwidth allocated to the instance. However, as VBP is meant to be design as a coarse-grained, easily implementable pricing model, we limit the scope of VBP to the case where each VM instance only support a single CoS, whether it is BE, DS, or MRG. Moreover, all the flows emitted from the instance gain equal share of network bandwidth, i.e., $(\theta_i, w_i) = (\theta/n_v, w/n_v)$, $\forall i$. VBP is a static pricing model, allowing the network bandwidth to be purchased as an integral part of a VM instance. However, a major drawback of VBP is that the aggregated bandwidth is statically pre-allocated and cannot be quickly adjusted to respond to network resource demand changes.

To address the above drawback of VBP, we also propose FBP. In this model, a customer pays an initial purchase fee for the default BE CoS as an integral part of a VM instance and then pays the DS and MRG CoSes on a per-flow-basis on demand. It also allows dynamic runtime service upgrading or downgrading by changing the pair of parameters and the corresponding price. FBP is more flexible than VBP, but is harder to implement and manage. In what follows, we propose pricing structures for the two models, separately.

A. VBP: VM-Based Pricing model

We propose to use the following pricing structure for VBP corresponding to the three CoSes.

BE CoS: This is the default CoS with (θ, w)=(0, 1). The price, P_{BE}, for this CoS may be set at, $P_{BE} = P_0$, where P_0 is a fixed basic price per unit time.

DS CoS: For this CoS, $(\theta, w) = (0, w > 1)$. The price, P_{DS}, for this CoS can be modeled as $P_{DS} = P_0 + P_1(w-1)$, where P_1 is a price per unit time. As we will show in the next section, a DS flow with $w > 1$ usually to be consistently smaller than the optimal one (i.e., w times of the BE flow rate)(will be explained later). All our results suggest that the average rate of DS flows with w=2 is about 1.6 to 1.7 times of the BE flow rate at high network load, and hence P_1 may be set at $0.6P_0$ to ensure that the flow rate is indeed proportional to the price paid.

MRG CoS: The price, P_{MRG}, for the MRG CoS may be formulated as $P_{MRG} = P_0 + P_1(w-1) + P_2\theta$, where P_2 is a price per unit data, in association with the minimum guaranteed rate.

Clearly, with VBP, in addition to CPU speed and memory size, a CoS with a specific (θ, w) pair can now be included for price tagging a VM instance. For example, a user may want to purchase VM instances with MRG CoS. Based on the application characteristics and an expected number of concurrently active flows, VM instances with a pair of parameters (θ, w) that matches the demand may be purchased, and making the performances of VM instances proportional to their prices paid.

B. FBP: Flow-Based Pricing model

In FBP, a customer is charged upfront for the use of the BE CoS. However, to use DS or MRG CoS, the customer will incur an additional charge, according to the specific values of the pair of parameters (θ, w) chosen for the flow. The additional charge, P^s, may follow a similar pricing structure as for the MRG CoS in VBP, i.e., $P^s = P_0^s + P_1^s(w-1) + P_2^s\theta$. Here P_0^s, P_1^s and P^s are the price per unit time and P_2^s are the price per unit of data sent.

Since our focus in this paper is on the price versus performance consistency, how to determine the parameters

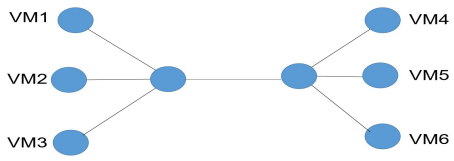


Figure 1: An example network.

in the pricing structures, i.e., $P_0(P_0^s)$, $P_1(P_1^s)$, $P_2(P_2^s)$, to balance the profit and user satisfaction is subject to future investigation.

C. Rate allocation Examples

Now we use a simple network topology presented in Figure 1[2] to illustrate how PACCP allocates flow rates under the two pricing models. Data flows are sent between three source-destination pairs, VM$_i$-VM$_{i+3}$ (i=1, 2 and 3), through a shared bottleneck link with bandwidth B. Assume that the propagation delays are the same for all the flows. In the follows, we discuss how the rates are allocated by applying VBP model and FBP model, respectively.

VBP : Assume that VM$_i$ (i=1,2 and 3) has n_i flows sending to VM$_{i+3}$, respectively. We first examine the case where all the flows are of BE CoS. In this case, the optimal rate allocation is such that each flow emitted from VM$_i$ receives $B/3n_i$ allocated bandwidth, respectively. Specifically, the shared bandwidth is first equally allocated to the three VMs with $B/3$ each, which is further equally allocated to each flow in a VM. For example, let n_1=1, n_2=2, and n_3=3. Then a BE flow from VM1, VM2, or VM3 is allocated $B/3$, $B/6$ or $B/9$ bandwidth, respectively.

Now we consider the case where there are two types of VMs running either BE or DS flows. Specifically, assume that both VM1 and VM2 run BE flows, and VM3 has DS flows with the pair of parameters (0, w). In this case, the optimal rate allocation is such that each BE flow from VM1 and VM2 is allocated the bandwidth of $B/((2+w)n_1)$ and $B/((2+w)n_2)$, respectively, and each DS flow gets $wB/((2+w)n_3)$. For example, suppose that n_1=1, n_2=n_3=2 and w=2, then each BE flow from VM1 and VM2 is allocated $B/4$ and $B/8$, respectively, and each DS flow from VM3 gets $B/4$.

Finally we exam the case with the presence of all three CoSes. Specifically, assume that VM1, VM2, and VM3 have BE flows, DS flows with $(0, w_1)$, and MRG flows with (θ, w_2), respectively. The optimal rate allocation is then to first allocate θ rate to VM3, and then proportionally allocates the remaining bandwidth to the three VMs to maximize the total utility. If θ =0, VM3 would be allocated a bandwidth of $Bw_2/(1+w_1+w_2)$. Otherwise, it would receive at least θ. As a result, VM3 gets $B_3 = max\{\theta, Bw_2/(1+w_1+w_2)\}$. Hence each MRG flow gets B_3/n_3, assuming that the minimum guaranteed rate is evenly assigned to each MRG flow. Then the remaining bandwidth $B_{BD} = B - B_3$ are allocated to VM1 and VM2, and hence VM1 and VM2 receive, $B_1 = B_{BD}/(1+w_1)$ and $B_2 = w_1 B_{BD}/(1+w_1)$, respectively, with each BE flow gets bandwidth B_1/n_1 and each DS flow gets B_2/n_2.

FBP : For this model, the optimal flow rate allocation for each flow is independent of the VM instance the flow is originated from. More specifically, assume that there are n_i^{BE}, n_i^{DS} and n_i^{MRG} BE, DS and MRG flows emitted from VMi (i=1, 2, and 3). Also assume that the pairs of parameters for all the flows belonging to the same CoS are the same. Namely, for BE, DS and MRG flows, they are (0,1), (0, w_1) and (θ, w_2), respectively. Also let, $n^{BE} = \sum_{i=1}^{3} n_i^{BE}$, $n^{DS} = \sum_{i=1}^{3} n_i^{DS}$ and $n^{MRG} = \sum_{i=1}^{3} n_i^{MRG}$. Then, the optimal flow rate allocation for flows from different VMs is dependent on n^{BE}, n^{DS}, n^{MRG} and CoSes only, not from which VMs they come from.

The optimal rate allocation is to first satisfy the minimum rate, θ, for all n^{MRG} MRG flows, and then allocates the remaining bandwidth to BE, DS and MRG flows in proportional to their weight values. Specifically, an MRG flow gets B_{MRG}=$max(\theta, Bw_2/(n^{BE}+w_1 n^{DS}+w_2 n^{MRG}))$. Then the remaining bandwidth $B_{BD} = B - n^{MRG} B_{MRG}$ is allocated to BE and DS flows with each BE flow getting $B_{BD}/(n^{BE}+w_1 n^{DS})$ and each DS flow getting $w_1 B_{BD}/(n^{BE}+w_1 n^{DS})$.

For example, assume that n_i^{BE}=n_i^{DS}=n_i^{MRG}=1 (i=1,2 and 3) and w_1=w_2=2. We first assume that $\theta = B/6$. In this case, B_{MRG}=$max(B/6, 2B/15)$=$B/6$, and then $B_{BD} = B/2$, so the optimal rate allocation is $B/18$ for a BE flow, $B/9$ for a DS flow and $B/6$ for a MRG flow. Now assume that $\theta = B/20$. In this case, B_{MRG}=$max(B/20, 2B/15)$=$2B/15$, then $B_{BD} = 3B/5$. So the optimal rate allocations are $B/15$ for a BE flow, $2B/15$ for a DS flow and $2B/15$ for a MRG flow.

The power of PACCP lies in the fact that with the right assignment of the pair (θ, w) in PACCP for each flow, the congestion control is automatically enabling users' prices into rate allocation and leads to the optimal price-aware rate allocation for any network topology without bandwidth reservation. Hence PACCP is readily to be implemented in today's datacenters for charging the network resource usage.

D. Implementation issue

PACCP is a price-aware congestion control protocol. It can be easily implemented in Linux kernel. The pair of parameters (θ, w) can be passed from the user space to the Kernel space through some standard device control

[2]This topology is different from the leaf-spine datacenter topology. However, if a datacenter has single bottleneck link at a time, the leaf-spine topology can be modeled as this topology.

functions, e.g., $ioctl()$. If the operating system (OS) is managed by the cloud service provider, the price charge can be directly executed by setting up/monitoring the pair of parameters passed from the user space to the Linux kernel. If the OS is administrated by tenants, the cloud service providers can move the congestion control from a data path to a congestion control plane [25], [26] or a virtual switch [27] which can be implemented in the network interface cards, and the price is charged based on the pair of parameters used in the congestion controllers.

V. Performance Evaluation

In this section, we first examine the optimality of PACCP by simulation as well as a small testbed implementation, and then test the price-performance consistency of PACCP for both pricing models in a large datacenter based on real-world workloads.

A. Optimality test by simulation

We first test PACCP in terms of the user utility maximization and optimal rate allocation by simulation. It is based on an event-driven simulator we developed. A 6x5 leaf-spine network topology with each rack having 40 hosts[3] is used. The bandwidth/propagation delay is set at 10Gbps /10 μs between a host and a leaf node and 40Gbps/20 μs between a leaf node and a spine node. The queue sizes for the 10/40 Gbps links are set at 150/450 kbytes (i.e., 100/400 packets). Suppose that in each rack, there are 20, 10 and 10 hosts running BE, DS and MRG flows, respectively.

We first consider a simple case, i.e., each host has one outgoing flow and one incoming flow. So there are a total number of 40 outgoing flows (20 BE, 10 DS and 10 MRG flows) in each rack, with 8 of them (i.e., 4 BE, 2 DS, and 2 MRG flows) going to exactly one of the other 5 racks. To test the optimality of PACCP, we assume that all the flows are extremely long-lived with unlimited amount of data to send. With this setup, 40 flows from each rack share a total of 200 Gbps (i.e., five 40 Gbps links) outgoing bandwidth. We first set the pairs of parameters to be (0, 1), (0, 2) and (2Gbps, 1) for BE, DS and MRG flows, respectively. In this case, the optimal flow rate allocation for each 40 Gbps leaf-spine link are 4 Gbps, 8 Gbps and 4 Gbps for each of the 4 BE, 2 DS and 2 MRG flows, respectively. We also consider another case where the only difference from the previous case is that the pair of parameters for MRG flows is changed to (7Gbps, 2). In this case, the optimal flow rates for each of the BE, DS and MRG flows are 3.25, 6.5 and 7 Gbps, respectively. Since for both cases, each VM only sources one flow, the flow rate allocations are the same for both VBP and FBP.

[3]In datacenters, a host can host one or multiple VMs. For simplicity, we assume that each host runs a single VM. Hence, VM and host are used exchangeably in the rest of the paper.

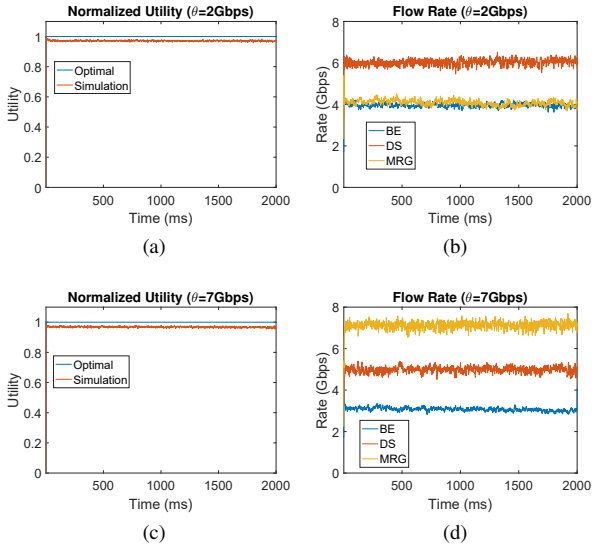
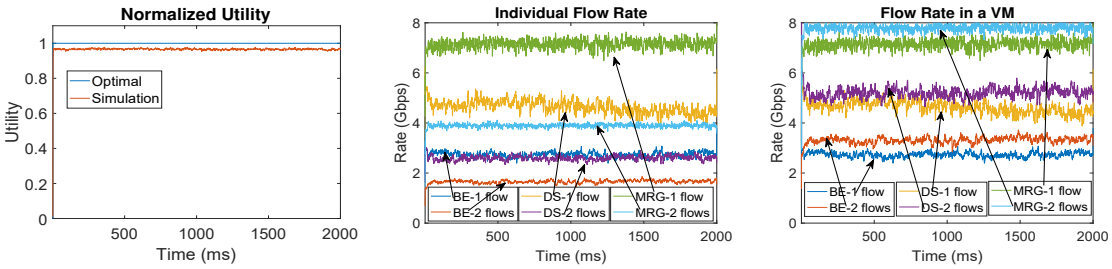


Figure 2: Normalized utility and rate allocation for each host having one flow: (a) and (b) θ =2 Gbps; (c) and (d) θ =7 Gbps.

Figures 2 (a) and (c) show the sum of user utility from simulation and the optimal one, V in Eq. (4), normalized to one. As we can see, the simulated one closely matches with, but is slightly lower than the optimal one for both cases. The reason why it is always lower than the optimal one is that for any transport congestion control protocol including PACCP, the aggregate flow rate cannot achieve 100% link utilization all the time, due to congestion feedback delay and finite granular control.

The rates of the three CoS flows, each averaged over all the flows in the same CoS, are depicted in Figures 2 (b) and (d). The average flow rates over time for BE, DS and MRG flows are 3.72/6.21/4.09 Gbps and 2.98/5.08/7.14 Gbps for the cases of θ =2 Gbps and 7 Gbps, respectively. The results indicate that the rates of MRG flows are always above the minimum guaranteed rate θ. The rate ratio between DS and BE is about 1.67/1.71, smaller than the optimal ratio 2, for both cases. This is because the optimal ones are obtained based on the assumption that the PACCP controllers for both BE and DS flows sharing the same bottleneck links will sense the congestion simultaneously. In practice, however, a flow of higher rate may sense more congestions than a flow of lower rate, and hence DS flows will incur more rate reduction. This explains why the flow rate ratio of the DS and BE flows is less than the optimal one.

To further test VBP, we create two types of hosts by adding one additional outgoing flow to each of the 10 BE, 5 DS and 5 MRG hosts in each rack, forming a second type of hosts, leaving the other half of hosts in each rack unchanged. As a result, each of this type of hosts now has 2 outgoing



(a) (b) (c)

Figure 3: Normalized utility and rate allocation for half hosts having one flow and half hosts having two flows : (a) Normalized utility; (b) Average flow rate; and (c) Total per VM rate.

flows. The flows generated from this type of hosts are called BE-2, DS-2 and MRG-2 flows, and the flows generated from the other hosts are denoted as BE-1, DS-1 and MRG-1 flows.

Now we consider the pairs of parameters (0, 1), (0, 2) and (7 Gbps, 2) for BE, DS and MRG hosts, respectively. This means that the pairs of parameters for BE-1, DS-1 and MRG-1 flows are (0,1), (0,2) and (7Gbps, 2), respectively, and the pairs of parameters for BE-2, DS-2 and MRG-2 flows are (0, 0.5), (0,1) and (3.5 Gbps, 1), respectively. As a result, the optimal flow rate allocation is 3.25 Gbps, 6.5 Gbps and 7 Gbps for BE-1, DS-1 and MRG-1 flows, respectively, and 1.625 Gps, 3.25 Gps and 3.5 Gbps for BE-2, DS-2 and MRG-2 flows, respectively.

Figure 3 (a) shows the results for the normalized user utility. Again, the simulated one is very close to the optimal one. Figure 3 (b) presents the simulated flow rates of individual types and CoSes. The average flow rates for BE-1/2, DS-1/2 and MRG1/2 are found to be 2.74/1.6, 4.65/2.66 and 7.16/3.85, respectively. Similar to the previous case, the rates of MRG flows are always above their required minimum rates and the flow rate ratios between DS-1 and BE-1, and DS-2 and BE-2 flows are about 1.66 and 1.7, lower than the optimal value of 2. The flow rate ratio between BE-1 and BE-2, DS-1 and DS-2, and MRG-1 and MRG-2 are 1.65, 1.78 and 1.86, respectively, also lower than the optimal value of 2, for the same reason explained earlier.

For the current case and under VBP, flows of the same CoS and from different VMs should be allocated the same total rate. For example, a DS-1 flow originated from one host should receive the same flow rate as the sum of two DS-2 flows originated from another host. Figure 3 (c) shows the testing results for the average flow rates originated from different types of hosts. We can see that the average flow rates from the two types of hosts of the same CoS are reasonably close to each other, with the rates from type 2 slightly higher than that of type 1. This is caused by the fact that each of the two flow from a type 2 host has a smaller flow rate than that of a flow from a type 1 host, and hence they sense less congestion signals, as explained earlier.

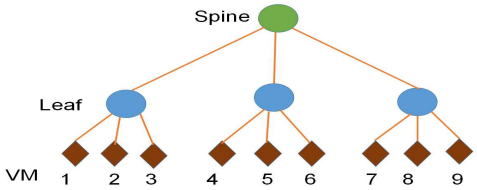


Figure 4: A 3x1 spine-leaf testbed.

B. *Optimality test in a real testbed*

We implement our PACCP in the Linux kernel. Our Linux code is modified based on the TCP Reno. In PACCP, the minimum guaranteed rate θ and the flow utility weight w are passed from the user space to the Kernel space through the standard device control function, $ioctl()$. A 3x1 leaf-spine datacenter network topology as shown in Figure 4 is set up using four Dell N4032F switches. Each link has 1 Gbps bandwidth. VMs 1, 4 and 7 are BE hosts, VMs 2, 5 and 8 are DS hosts and VMs 3, 6 and 9 are MRG hosts. Each VM originates 1 long-lived flow. Hence the flow rate allocation is the same, regardless whether VBP or FBP is in use. The destinations of VM i are $(i+3)\%9$ (for i=1, 2, ..., 9). The pairs of parameters are set at (0,1), (0, 2) and (400Mbps, 2) for BE, DS and MRG flows, respectively. With this setup, the optimal flow rates are 200 Mbps, 400 Mbps and 400 Mbps for BE, DS and MRG flows, respectively.

Figure 5 shows the average flow rates for flows of the three CoSes. The average rate of MRG flows is about 410 Mbps, above the minimum guaranteed rate 400 Mbps. The average rate of DS and BE flows are about 310 Mbps and 180 Mbps, respectively, resulting in a flow ratio of about 1.7, less than the optimal ratio 2. These results agree with the simulation results.

In summary, both simulation and testbed testing results indicate that with PACCP, MRG flows have high chance to meet their minimum guaranteed rates. Moreover, the DS flows can indeed gain more bandwidths when they compete with BE flows, which however, are consistently lower than

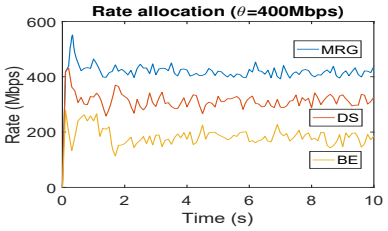


Figure 5: Rate allocation in the testbed.

the optimal ones. This implies that the pricing parameters for DS flows in both VBP and FBP need to be adjusted to reflect the biased relative flow rate. We found that a DS flow with w=2 achieves about 1.6 times the flow rate as a BE flow at high load. Although more detailed study at higher weight values are warranted, this observation does suggest that one may set P_1 (P_1^s) at some smaller value than P_0(P_0^s), e.g., 0.6 P_0 (P_0^s) for VBP (FBP). Note that DS is meant to outperform BE at high load. At low load, they offer similar performance. Hence, DS related pricing must be determined at the high load.

C. Performance evaluation with real datacenter workloads

In datacenters, the flow size varies significantly [17], [11], [28]. While most of the flows are short-lived, having flow size less than 100K, a small percentage of long-lived, big flows consume most of the network bandwidth. In the following, the flow allocations using PACCP with the two pricing models are tested using real datacenter traffic workloads, i.e., Websearch [11] and Data-mining [28], as input. The focus is placed on the testing of the price-performance consistency, i.e., whether the flow rate allocated by PACCP matches the expected rate allocation based on the two pricing models.

We use the same network setup as the previous one, i.e., a 6x5 leaf-spine network topology with the same link bandwidths and with each rack having 40 hosts. The flows are dynamically generated, following a Poisson process. The average flow arrival interval is used as a tunning knob to set the network load at desired levels. When a flow arrives, a source host is randomly selected and then a destination host is randomly selected from a different rack.

The average flow completion time (FCT) is used as a performance metric, which is equivalent to the average flow rate, as the flow sizes for all flows are given. The overall FCT and FCTs for small flows (with size less than 100K bytes), medium flows (with size between 100 Kbytes and 1 Mbytes) and big flows (with size larger than 1 Mbytes) are measured. For MRG flows, a minimum guaranteed rate and hence, a flow deadline is set. Another performance metric used is the flow deadline meet ratio (DMR), which is applied to the overall flows, as well as the small flows, medium flows, and big flows, separately. Although BE and DS flows are deadline unaware, we compare the DMRs for flows with deadline using the BE and DS services against that using the MRG service to reveal how much MRG can help improve DMR over the other two.

1) VBP: We first examine the performance of PACCP with VBP. Consider the case where there are 20, 10 and 10 hosts in each rack running BE, DS and MRG flows, respectively. The pairs of parameters are set at (0, 1), (0,2) and (5 Gbps, 2) for BE, DS and MRG flows, respectively. The flow deadline for each of n_a active outgoing MRG flows at a host is set at the flow size divided by 5/n_a Gbps. We assume that the flow deadline is lower bounded at 1 ms, as the PACCP connection setup time is taken into account.

We run PACCP using the Websearch workload [11]. Figures 6 (a) and (b) present the average FCTs for the overall, small, medium and big flows (normalized to the FCT for the BE flows). We see that both DS and MRG flows indeed perform better than BE flows for all load cases. For small and medium DS/MRG flows, their FCTs are less than 0.8 times (i.e., the flow rates are 1.25 times higher) of BE flows at all load cases. As these flows are short-lived flows and may be completed before they reach their optimal allocated flow rate, the performance gains for these flows come from the faster rate increase with the help of w and r_{cos} (i.e., θ). The results indicate that PACCP is really effective for short-lived flows to enabling price based rate allocation. At light loads, the difference for big flows is small, about 0.9 times of that for the BE flows. This is because at light loads, there is enough bandwidth to accommodate the desired user utilities for all the individual flows of different CoSes, which hardly need to compete against one another for the network resource. Hence long flows have enough time to fully explore the available bandwidth, making the small performance gains.

The performance gains increase quickly as the network load increases. In the high load region (e.g., at 80%), the FCTs for the overall/small/medium/big DS and MRG flows are about 0.62/0.62/0.6/0.63 and 0.61/0.62/0.58/0.62 times of BE flows, respectively. In other words, the average flow rate allocated to DS flows versus BE flows is about 1.6 and 1.7 times, lower than the optimal ratio of 2, which agrees with the findings for the previous long-lived flow cases. MRG flows perform slightly better than DS flows for all cases. The performance gains are about 5% for small and medium flows, but very little for the overall and big flows. The close performance between the DS and MRG flows arises because both DS and MRG have the same weight value of 2. Hence, they are expected to receive equal flow rate allocation, provided that the minimum guaranteed rates for MRG flows are satisfied, which is indeed the case. The fact that MRG flows perform slightly better is because MRG flows open up their send windows faster than DS flows until the flow rates reach their minimum guaranteed rates.

Figures 6 (c) and (d) show the DMRs for the overall,

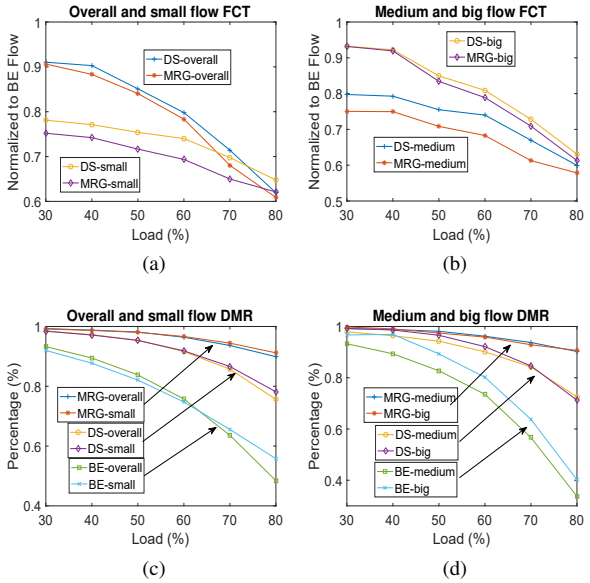


Figure 6: VBP with Websearch workload: (a) overall and small flow FCT; (b) Medium and big flow FCT; (c) Overall and small flow DMR; and (d) Medium and big flow DMR.

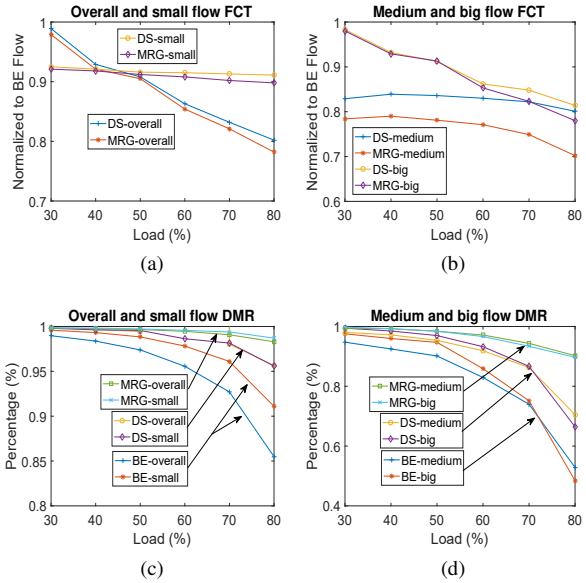


Figure 7: FBP with Data-ming workload (a) overall and small flow FCT; (b) Medium and big flow FCT; (c) Overall and small flow DMR; and (d) Medium and big flow DMR.

small, medium and big flows. The DMRs for MRG flows are always higher than BE and DS flows. The overall DMR for MRG flows is above 90% even at high load, whereas the DMR for BE flows is below 50%. While the DMRs for medium and big MRG flows are above 90%, the corresponding DMRs for DS and BE flows drop below 80% and 50%, respectively. This clearly demonstrates the importance of MRG in providing high probability of meeting flow deadlines.

2) FBP: Finally, we evaluate the performance of PACCP with FBP. For FBP, a customer can run flows of different CoSes in a VM. In our simulation, an outgoing flow from a host has 60% chance to be a BE flow, 20% chance to be a DS flow, and 20% chance to be an MRG flow. The pair of parameters for BE, DS, and MRG flows are set at (0,1), (0, 2) and (2.5 Gbps, 2), respectively. In FBP, a VM may have all the three types of flows at the same time.

Figure 7 gives the results using the Data-mining workload as input. Overall, the results are similar to those for VBP with Websearch workload. But the performance gains, in terms of both FCTs and DMRs for DS and MRG flows at high loads, are less than those for VBP. The FCT gain for small flow is almost a constant for all load cases for the following reason. Most of the small flows in the Data-ming workload are composed of only a few packets (about 40% flows have a single packet and about 80% flows have no more than 6 packets), which finish in just one or two RTTs, and hence the fast rate increase has limited effect on these flows. For VBP, the utility weight w for a host is the sum of the weights for all the outgoing flows from that host, meaning that smaller weights are assigned to individual flows for the VBP case than those for the FBP case. It means that the flow increase rate in VBP is slower than that in FBP, and hence the total number of congestions in VBP is less than that in FBP. Hence the different congestions sensed between two flows with different rates are reduced and hence the performance gains in VBP are greater than that in FBP.

The above results demonstrate that the proposed PACCP can indeed enable price-aware flow rate allocation in cloud, particularly for short-lived flows, including soft minimum rate guarantee and relative additional rate allocation, commensurate with two pricing models. PACCP is a fully distributed control protocol and only needs software upgrading in the end hosts and does not involve any core switches, and hence it is readily to be implemented in current datacenters.

VI. Related Work

Network price modeling for datacenters is a heavily studied subject [1], [2], [3], [4], [5], [7], [8], [29], [30], [31]. Some of them [3], [29], [31] focus on the study of the economical impact of cloud resource pricing on maximizing the revenue. The other schemes [4], [30] focus on how to enable flow pricing in proportional to flow rate allocation through bandwidth reservation. The price is usually dynamically generated either through an auction process or a time-varying price table. However, the bandwidth reservation generally incurs significant overheads and hence are not suitable for today's datacenter applications involving

bursts of short-lived flows. As a distributed network pricing solution, Softbw [1] can schedule flows at the source hosts and hence can deal with short-lived flows. However, it only works for congestion free datacenter networks.

Although price-agnostic, many solutions have been proposed to improve datacenter flow rate allocation and provide fair bandwidth sharing and/or minimum rate guarantee. Since such solutions may lead to effective pricing models, in what follows, we briefly review some of such solutions.

First, some congestion control protocols with minimum assistance from in-network nodes [11], [13], [32], [33] are proposed to improve the performance of datacenter applications. Some of them [11], [32], [33] are focused on improving average throughput, their congestion controls are based on explicit congestion notification (ECN) or queuing delay, or on the flowlet granularity. D^2TCP provides deadline-aware [13] flow rate allocation through ECN. However they cannot provide provable fairness and minimum flow rate or flow-deadline guarantee and multiple CoSes, as is the case for PACCP. Second, the Hose and Pipe models are widely used for the design of bandwidth allocation schemes [2], [34], [35], [36], [37], [38], [39] for datacenter applications. In these schemes, all the VMs are connected to a central (virtual) switch by a dedicated link (hose) for traffic control and minimum bandwidth guarantee. Oktopus [34] and SecondNet [35] support bandwidth guarantee through static reservation. Seawall [37] and NetShare[38] uses flow weight or per-tenant weight for TCP-like flows to achieve max-min fairness. Gatekeeper [40] uses a hypervisor-based mechanism for bandwidth reservation for bisection-bandwidth networks. Tag [39] uses a tenant application graph to more accurately predict the bandwidth demand and hence more effective allocate bandwidth for applications with heterogeneous bandwidth demands. As the bandwidth reservation is not integrated with the congestion control protocols, these schemes cannot allocate flow rates in a work-conserving manner, hence wasting network resources.

The ability to support multiple CoSes and directly enforce flow rate allocation in congestion control makes PACCP a highly resource-effective, work-conserving solution. Moreover, to the best of our knowledge, it is the first price-aware transport congestion control protocol purposely design for cloud applications, and can be readily deployed to current datacenters.

VII. Conclusions

In this paper, we propose PACCP, a price-aware congestion control protocol for cloud applications. PACCP is a NUM based optimal congestion control protocol and supports multiple CoSes, including best-effort service (BE), differentiated service (DS) and minimum rate guaranteed (MRG) service. PACCP seamlessly integrates congestion control with two pricing models, a coarse-grained VM-Based Pricing model (VPB) and a fine-grained Flow-Based Pricing model (FBP). The flow rate allocated by PACCP is determined by a pair of parameters, i.e., a minimum guaranteed rate and a utility weight, which are, in turn, determined by the paid price. PACCP is evaluated by both large scale simulation and small testbed implementation. The experimental results demonstrate that PACCP can indeed achieve high probability of providing minimum rate guarantee, high bandwidth utilization and proportional flow rate allocation, commensurate with the two pricing models.

Acknowledgment

This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and the US NSF under Grant No. CCF XPS-1629625, CCF SHF-1704504 and CCF SHF-2008835.

References

[1] J. Guo, F. Liu, T. Wang, Tao and J. C. Lui, John, *Pricing Intra-datacenter Networks with Over-Committed Bandwidth Guarantee*, USENIX ATC, 2017.

[2] L. Popa, G. Kumar, M. howdhury, A. Krishnamurthy, S. Ratnasamy and I. Stoica, *FairCloud: Sharing the Network in Cloud Computing*, ACM SIGCOMM, 2012.

[3] D. Niu, C. Feng and B. Li, *Pricing cloud bandwidth reservations under demand uncertainty*, ACM SIGMETRICS performance evaluation review, 2012.

[4] H. Ballani, K. Jang, T. Karagiannis, C. Kim, D. Gunawardenam and G. O'Shea, *Chatty tenants and the cloud network sharing problem*, Usenix NSDI, 2013.

[5] H. Shen and Z. Li, *New Bandwidth Sharing and Pricing Policies to Achieve A Win-Win Situation for Cloud Provider and Tenants*, IEEE INFOCOM, 2014.

[6] V. Jalaparti V, I. Bliznets and S. Kandula, *Dynamic pricing and traffic engineering for timely inter-datacenter transfers*, ACM SIGCOMM, 2016.

[7] A. Jin, W. Song, P. Wang, D. Niyato and P. Ju, *Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing*, IEEE Transactions on Services Computing, v9(6), pp895-909, 2016.

[8] A. Jin, W. Song and W. Zhuang, *Auction-based resource allocation for sharing cloudlets in mobile cloud computing*, IEEE Transactions on Emerging Topics in Computing, v6(1), pp45-57, 2018.

[9] S. Kansal, H. Kumar, S. Kaushal and A.K. Sangaiah, *Genetic algorithm-based cost minimization pricing model for on-demand IaaS cloud service*, The Journal of Supercomputing, v76, pp1536-1561, 2020.

[10] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K.K. Ramakrishnan, *Optimal content placement for a large-scale VoD system*, IEEE/ACM Transactions on Networking, v24(6), pp2114-2127, 2016.

[11] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta and M. Sridharan, *Data center TCP (DCTCP)*, ACM SIGCOMM, 2010.

[12] M. Andrews, *Searching the Internet*, IEEE Software, v29(2), pp13-16, 2012.

[13] B. Vamana, J. Hasan and T.N. Vijakumar, *Deadline-Aware Datacenter TCP (D^2TCP)*, ACM SIGCOMM, 2012.

[14] A. Roy, H. Zeng, J. Bagga, G. Porter, and A.C. Snoeren, *Inside the Social Network's (Datacenter) Network*, ACM SIGCOMM, 2015.

[15] J. Dean and L.D. Barroso, *The Tail at Scale*, Communications of the ACM, v56, pp74-80, 2013.

[16] C. Wilson, H. Ballani, T. Karagiannis and A. Rowstron, *Better Never than Late: Meeting Deadlines in Datacenter Networks*, ACM SIGCOMM, 2011.

[17] T. Besn, A. Akella and D.A. Malta, *Network Traffic Characteristics of Data Centers in the Wild*, ACM SIGCOMM Conference on Internet Measurement, 2010.

[18] Z. Wang and A. Singhal, Y. Wu, C. Zhang, H. Che, H. Jiang, B. Liu and C.M. Lagoa, *HOLNET: A holistic traffic control framework for datacente networks*, IEEE ICNP, 2020.

[19] L. Ye, Z. Wang, H. Che, Hao and C. Lagoa, *TERSE:A Unified End-to-End Traffic Control Mechanism to Enable Elastic, Delay Adaptive, and Rate Adaptive Services*, IEEE Journal on Selected Areas in Communications, v29(5), pp938-950, 2011.

[20] L. Ye, Z. Wang, H. Che and H.B.C. Chan and C.M. Lagoa, Constantino M, *Utility function of TCP*, Computer Communications, v32(5), PP800-805, 2009.

[21] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, 2014.

[22] B.A. Movsichoff, C. Lagoa and H. Che, *End-to-End Optimal Algorithm for Integrated QoS, Traffic Engineering, and Failure Recovery*, IEEE Transactions on Networking, v15(4), pp813-823, 2007.

[23] E. Kohler, M. Handley and S. Floyd, *Designing DCCP: congestion control without reliabilit*, ACM SIGCOMM, 2006.

[24] L. Ye, Lei and Z. Wang, *A QoS-aware congestion control mechanisim for DCCP*, IEEE Symposium on Computers and Communications (ISCC), 2009.

[25] A. Narayan, F. Cangialosi, P. Goyal, S. Narayan, M. Alizadeh and H. Balakrishnan, *The Case for Moving Congestion Control Out of the Datapath*, ACM HetNets, 2017.

[26] A. Kaufmann, T. Stamler,S. Peter,N.K. Sharma, A. Krishnamurthy and T. Anderson, *TAS: TCP Acceleration as an OS Service*, EuroSys, 2019.

[27] V. Jeyakumar, M. Alizadeh, D. Mazieres, B. Prabhakar, C. Kim and A. Greenberg, *Eyeq: practical network performance isolation at the edge*, USENIX NSDI, 2014.

[28] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim,and P. Lahiri, D.A. Maltz, P. Patel, and S. Sengupta, *VL2: A Scalable and Flexible Data Center Network*, ACM SIGCOMM, 2009.

[29] D. Niu and B. Li, *An efficient distributed algorithm for resource allocation in large-scale coupled systems*, IEEE INFOCOM, 2013.

[30] X. Wu, M. Liu, W. Dou, L. Gao and S. Yu, *A scalable and automatic mechanism for resource allocation in self-organizing cloud*, Peer-to-Peer Networking and Applications, v9(1), pp28-41, 2016.

[31] H. Wang, Q Jing, R. Chern, R. He and B. Qian, *Pricing cloud bandwidth reservations under demand uncertainty*, USENIX HotCloud, 2010.

[32] V. Arun and B. Hari, *Copa: Practical delay-based congestion control for the internet*, ACM NSDI, 2018.

[33] J. Perry, H. Balakrishnan and D. Shah, *Flowtune: Flowlet control for datacenter networks*, ACM NSDI, 2017.

[34] H. Ballano, P. Costa, T. Karagiannis and A. Rowstron, *Towards predictable datacenter networks*, ACM SIGCOMM, 2011.

[35] C. Guo, G. Lu, H.J. Wang, S. Yang, C. Kong, P. Sun, W. Wu andY. Zhang, *SecondNet: a data center network virtualization architecture with bandwidth guarantees*, ACM CoNext, 2015.

[36] L. Popa, P. Yalagandula, S. Banerjee, J.C. Mogul, Y. Turner and J.R. Santos, *ElasticSwitch: Practical*, ACM SIGCOMM,2013.

[37] A. Shieh, S. Kandula, A. Greenberg, C. Kim and B. Saha, *Sharing the Data Center Network*, USENIX NSDI, 2011.

[38] V.T. Lam, S. Radhakrishnan, R. Pan, A. Vahdat and G. Vargese, *Netshare and stochastic netshare: predictable bandwidth allocation for data centers*, ACM SIGSOMM Communications Review, v42(3), pp5-11, 2012.

[39] J. Lee, Y. Turner, M. Lee, L. Popa, S. Banerjee, J. Kang and P. Sharma, *Application-driven bandwidth guarantees in datacenters*, ACM SIGCOMM, 2014.

[40] H. Rodrigues, J.R. Santos, Y. Turner, P. Soares, Paolo and D. Guedes, *Gatekeeper: Supporting Bandwidth Guarantees for Multi-tenant Datacenter Networks*, USENIX WIOV, 2011.