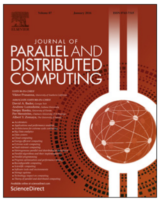
Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc



Deadline-aware rate allocation for IoT services in data center network

Bo Shen [a,b,*], Naveen Chilamkurti [c], Ru Wang [d], Xingshe Zhou [b], Shiwei Wang [e], Wen Ji [f]

[a] *Department of Electrical Engineering, Princeton University, NJ, USA*
[b] *School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China*
[c] *Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia*
[d] *College of Information Engineering, Northwest A&F University, Yangling, China*
[e] *Weihai Yuanhang Technology Development Co. Ltd., Weihai, China*
[f] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*

HIGHLIGHTS

- DRA algorithm is proposed, which is an abstract model to schedule flows in DCNs.
- A non-cooperative game is used to describe the interaction involving the allocation.
- DRA is a preemptive algorithm and guarantees the utility of DCNs.
- DRA shows good real-time performance, while it achieves fairness and shorter waiting time.

ARTICLE INFO

Article history:
Received 29 March 2017
Received in revised form 22 August 2017
Accepted 22 September 2017
Available online xxxx

Keywords:
Data center network
Big data
Online service
Internet of things
Incast congestion

ABSTRACT

Data center is the key infrastructure for a plenty of applications involving a high volume of data in Internet of Things (IoT). The Data Center Network (DCN) connecting multiple servers plays a vital role. Its mechanism for managing the traffic has a significant impact on the performance of IoT services. To guarantee the real-time performance of IoT services is one of the major challenges. In the paper Deadline-aware Rate Allocation (DRA) algorithm for scheduling the heterogeneous flows in DCNs is proposed. A non-cooperative game-theoretic framework is introduced to model the interactions in the scenario. The core idea of DRA is to assign the traffic with deadline constraints a higher priority. The worker with a lower served rate in the past period is assigned a higher priority. Meanwhile, DRA is a kind of preemptive algorithm. Simulation results have shown that under the mechanism flows wait shorter time and the real-time performance is guaranteed. DRA also achieves good fairness among different IoT services.

1. Introduction

The past few years have been witnessed an unprecedented proliferation of data space. Internet of Things (IoT), which connects many objects in the real-world, must be one important sources [7,21,9,34]. The technologies contributing to the success of IoT include large-scale data analysis [14,11], wireless communication [30,28], data processing, information diffusion and fusion [43]. With the aid of IoT-related technologies, applications utilize a high volume of data to provide more accurate services. The amounts of data pose challenges on traditional computing paradigm. Cloud computing paradigm is proposed to satisfy the computing requirement in the era of big data [23,5]. It provides on-demand computational and storage resources to users. By using the new computing paradigm, infrastructure, platform and software are delivering as services [6]. Due to the complicated requirements of storage and computing, the data centers, containing a very large number of servers, are used to manage and process the big data. Supported by the data centers, various web-service providers like Google, Amazon, Microsoft and Facebook take a plenty of online services to users. These examples include web search, social networking [42], advertising [27], recommendation systems, and location-based services [13,12]. Both sides of the services, i.e. service providers and service users, can benefit to meet their demands. IoT and the new computing paradigm are attracting extensive attentions from both academia and industry since their births. IoT based on computing becomes more prevalent.

Based on the data acquired from IoT, the data centers provide services to a large number of end-users by various applications.

[*] Corresponding author at: School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China.

E-mail addresses: boshen@princeton.edu (B. Shen), N.Chilamkurti@latrobe.edu.au (N. Chilamkurti), ruwang@nwafu.edu.cn (R. Wang), zhouxs@nwpu.edu.cn (X. Zhou), wsw@wh-yuanhang.com (S. Wang), jiwen@ict.ac.cn (W. Ji).

https://doi.org/10.1016/j.jpdc.2017.09.012
0743-7315/© 2017 Elsevier Inc. All rights reserved.

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.

These users visit the data centers via various applications with heterogeneous Quality of Service (QoS) requirements. In order to complete the user's demands, the data centers translate the services into the computing jobs processed by the servers. The servers exchange the data related to the jobs for cooperation. Imposed by the Service Level Agreements (SLAs), data centers should send back the response to user's request within a specified latency. Latency is a critical metric in the data centers [2]. How fast the computing jobs can be done is important when the performance of a data center is evaluated [20]. If the delivery exceeds the specified deadline, the response is dropped. However, the correct results of applications may depend on the responses from multiple servers in data centers. The uncompleted or incorrect result due to response expiration can deteriorate end-user's experience [2,35]. How to meet the soft real-time constraints becomes one of the major challenges for such applications. The Data Center Network (DCN) connecting multiple servers plays a vital role [10]. It provides communication capacity in single data center [20,1,8]. Data exchange among multiple data centers also comes true with the aid of DCNs. Its properties and mechanisms for managing the traffic have a significant impact on the performance of big data analysis in data centers [41].

As the amounts of data continue exploding, the increasing traffic demand arises in data centers. The Partition/Aggregate workflow pattern is proposed for parallel processing in data centers. Many online applications follow this pattern. In the structure, the root aggregator partitions user's request into multiple tasks. Then these tasks are further partitioned into sub-tasks. The sub-tasks are completed by the workers. The workers send their responses to root aggregator via aggregators. There are several challenges for the data center networking when a high volume of traffic is introduced. First, the visiting from an end-user is non-deterministic. The data flow in DCNs changes dynamically due to uncertain dynamic of user's requests. According to the investigation in the flow characteristics, steady flows and burst flows coexist. The traditional static, off-line rate allocation algorithms may suffer from low efficiency due to the fluctuating traffic volumes. An effective DCNs protocol becomes a vital issue in IoT cloud computing. Second, the flows in data centers require heterogeneous QoS goals. The long flows require desired throughput, while the short flows and the query traffic want the timely transmissions. Meanwhile, the timing features, i.e. deadlines, vary according to the applications' differences. DCNs protocol should support various requirements of IoT applications. How to design an effective algorithm to manage flows should be carefully considered. To summarize, QoS goals of heterogeneous flows must be satisfied simultaneously. The resource provisioning techniques supporting differentiated services in a dynamical and adaptive manner are preferred. Third, incast phenomenon has been a severe problem in DCNs, which may lead to throughput collapse [31,48]. Incast happens when multiple flows from different sources aggregate into the same sink simultaneously. Overwhelmed packets arriving the switch overflow its buffer. The occurrence of severe packet loss increases the latency of flows. Then the performance degradation of data centers follows.

As an increasing number of IoT services requires on-line interactions, data centers have to provide real-time response. For this purpose, we aim to design a policy that can guarantee the real-time performance of network flows while it is easy to implement in data centers. Deadline-aware Rate Allocation (DRA) algorithm for scheduling the heterogeneous flows in DCNs is proposed. As the optimal algorithm to schedule flows for multiple links simultaneously does not exist [3], in the paper we focus on local optimization. DRA models the bandwidth allocation problem as a non-cooperative game to guarantee the flow deadlines. Non-cooperative game provides a framework to describe the interactions among multiple interest-conflicted competitors. In the framework the gain of equilibrium strategy derives each competitor to make its most rational decision. Compared with other game-theoretic frameworks, additional overhead introduced by non-cooperative model is small.

In order to maximize each link's payoff, the link needs to evaluate the price first. In the game the price is related to multiple factors of a flow. From the perspective of link-level, the key point of the policy is that DRA assigns higher priorities to the traffic with deadline constraints within a link. DRA also assigns higher priorities to the links with lower served rates in the past period if the candidate flows have the same deadline constraints. In addition, DRA explores a tradeoff between the deadlines and the historical statistics of each transmission link. Meanwhile, DRA belongs to the kind of preemptive algorithms. When a new flow arrives at a link, it can become the Head-Of-Line (HOL) flow of its link if it has the smallest deadline, even if the former HOL flow has been transmitted a part of its traffic. From the perspective of switch-level, the rate is allocated based on the descending order of valuated price vector. When the switch allocates the bandwidth among workers, the link that provides the highest evaluated price is considered first. The candidate link get its desired bandwidth or all the residual bandwidth. Then the bandwidth is allocated to the next link if there is spare capacity. The process is repeated until all the residual bandwidth is assigned. The results of extensive simulations have shown that the mechanism guarantees the real-time performance. The links also wait shorter time for successful transmissions. At the same time, DRA achieves a good fairness among links. The proposed strategy provides real-time communication support to cloud-related services constrained by heterogeneous deadlines. Therefore, DRA guarantees the upper-boundary of latency of the response flow. User's perceived service quality is well satisfied. The simulation results show that short latency with fair guarantee in data centers is achieved.

The key contributions of this paper can be concluded as follows:

- We design Deadline-aware Rate Allocation (DRA) algorithm, which is an efficient model to allocate available bandwidth with deadline constraints in data center networks. DRA can be applied to many resource allocation scenarios with the many-to-one feature.
- A non-cooperative game-theoretic framework is proposed to describe the interaction in the rate allocation. Then the paper derives a solution of bandwidth allocation by computing Nash Equilibrium point of the game.
- DRA is implemented in data center networks to schedule flows with preemptive mechanism via simulations. The experimental results show that DRA achieves good realtime performance.
- DRA also provides fair allocation among multiple competitors without any loss of real-time performance. Meanwhile, DRA guarantees that the flows wait shorter time in transmission links.

The remainder of this paper is organized as follows: Section 2 discusses the communication characteristics in DCNs, and then the state of the art of related works is briefly introduced. Section 3 describes the system model and notations used in the paper. Based on these, the non-cooperative game model and the rate allocation algorithm are presented in Section 4. Section 5 evaluates the performance of the proposed algorithm. Section 6 concludes the paper.

2. Background

In this section we briefly introduce some foundations of the data center networks, including the network topology, the traffic pattern and the state-of-the-art of related works.

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.

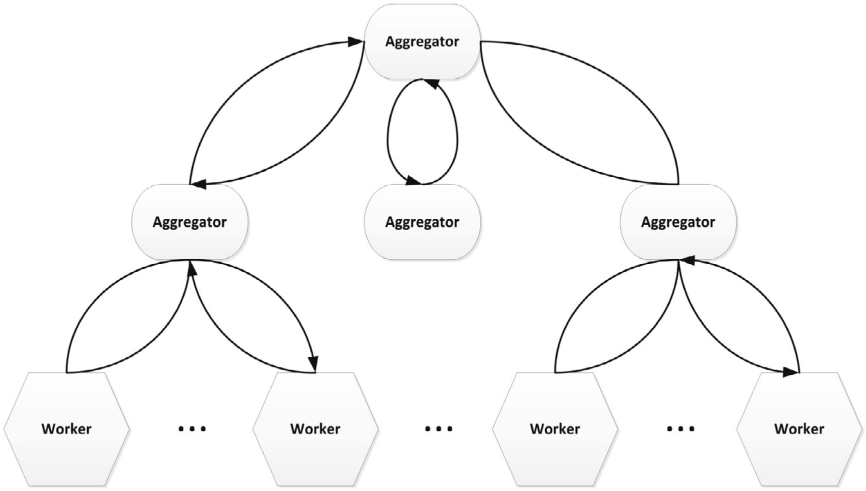


Fig. 1. The Partition/Aggregate design pattern in DCNs.

2.1. Data center communications

According to the measurements in [2], the applications in data centers contribute to three kinds of traffic: bursty query traffic, time-sensitive short flows and throughput-sensitive long flows. The query traffic is time-critical, following the Partition/Aggregate pattern. The size of this kind of traffic is very small, usually from 2 kB to 20 kB. The short flows and the long flows both belong to the background traffic. The short flows are used to update the control state on the workers with size from 50 kB to 1 MB. They are also time-critical flows. These time-critical traffic consists of web search requests, online gaming commands, etc. The long flows, with the size from 1 MB to 50 MB, are used to copy fresh data to the workers. Computing-intensive and data-intensive jobs also contribute to this kind of traffic. For the query traffic and the short flows associated with deadlines, the transmissions must be finished in a timely manner. The long flows must be transmitted to achieve the satisfactory throughput.

It is well known that the Partition/Aggregate design pattern is widely used in today's large-scale web services, which is shown in Fig. 1. The flow structure has been successfully applied to many web applications and data processing services, such as web search, social network, MapReduce, Dryad and advertisement recommendation systems [2,45,18]. When a user posts a request/task, it is send to the root server of the tree-like topology. The root server, also called Top-Level Aggregator (TLA), receives the request and then partitions it into several small pieces. These small pieces of requests are distributed to the servers at the intermediate level of the tree. The server in this level acts as Mid-Level Aggregator (MLA). MLAs further partition the small pieces again and distribute them to the servers at the leaf-level. The workers, i.e. the leaf-level servers, execute the final computing tasks. According to the computing result, each worker generates a response and sends it to the parent server. The MLA combines multiple responses from its child-workers for aggregation. Then MLA generates an aggregated response and delivers the response to the TLA. In a similar way, the TLA aggregates all the aggregated responses from the MLAs. The TLA makes the final response to user's request based on the aggregated result and sends back to the user.

The interactive latency of IoT service is a key metric related to Quality of Experience (QoE). It greatly affects the service provider revenue as well [25]. In typical web scenarios, when a TLA receives a request from an end-user, it should generate a response within a deadline. Each partitioned piece-task also partitions the deadline and inherits a part of the deadline. For example, if SLA requires the data center to send back a response within 100 ms, TLA may set the deadlines to 50 ms for MLAs to generate the responses. Then MLAs set the deadlines to 20 ms for workers. When a low-level node misses its deadline, the response flow is discarded. The high-level node continues to aggregate the computing result based on the existing responses. The uncompleted result reduces the quality of the response, potentially affecting the user's QoE and the provider's revenue. Therefore, to satisfy the response deadlines in data center communications is vital to provide the desired IoT service in DCNs.

Many-to-one traffic pattern is common in many important datacentre applications, such as MapReduce and web searching [46,32]. When multiple synchronized servers send data to the same aggregator simultaneously, the flows may exhaust the switch buffer, resulting in packet losses. Even if the flow size is small, highly bursty traffic from multiple sources can still overflow the switch buffer in a short time. Retransmission due to intense packet loss further deteriorates the congestion. Then the notorious throughput collapse happens. This phenomenon certainly leads to missing the response deadlines. Incast naturally arises from the Partition/Aggregate workflow pattern.

2.2. Related work

The issue of resource allocation exists in various networks [29]. There are a plenty of works aiming to construct efficient interconnecting protocols to support high volume of traffic and high-speed communication in DCNs. Based on the deadline factor, we classify these works into two categories: deadline-agnostic protocol and deadline-aware protocol.

Data Center TCP (DCTCP) proposed by Alizadeh et al. is a deadline-agnostic protocol. It is a kind of fair-shared protocol [2]. This means DCTCP aims to achieve a fair allocation and ignores the application deadlines. DCTCP uses the interpretation of Explicit Congestion Notification (ECN) message to modify the congestion window.

The Deadline-Driven Delivery control protocol, named D^3, is presented in [45]. The protocol gathers the exact information of each flow, such as deadline and size, to allocate the bandwidth. It is a deadline aware protocol. However, D^3 serves the flows in the First-Come-First-Serve (FCFS) manner. This mechanism may allocate the bandwidth to the flow with a bigger deadline rather than the flow with a smaller deadline. Furthermore, D^3 also requires the changes of hardware chips. It is hard to implement the protocol in the practical deployment.

As D^3 assigns the bandwidth in the FCFS manner, it may fail to schedule the flows with very short deadlines. Preemptive Distributed Quick (PDQ) is proposed to overcome this drawback [24]. PDQ uses the Earliest Deadline First (EDF) to minimize mean flow completion time and the Shortest Job First (SJF) to minimize the number of deadline-missing flows. PDQ supports the flow preemption and can suspend the contending flows. The protocol is implemented in a distributed style and allows multiple switches to cooperatively collect the network information.

Deadline-Aware Datacenter TCP (D^{2}TCP) is a deadline aware protocol with a fully distributed implementation [39]. D^{2}TCP inherits the core function of DCTCP. However, D^{2}TCP uses ECN message and deadline information to adjust the congestion window size via a gamma-correction function. The key idea behind D^{2}TCP is that a high priority is assigned to the flow with a smaller deadline. The bandwidth of each flow is determined proportional to the priority of the flow.

Deadline Aware Queue (DAQ) schedules the traffic in data center guaranteeing the tight deadline for the short flows and the transmission rate for the long flows [17]. At the supporting switches, DAQ uses three queues to differentiate flows: an urgent queue for the real-time flow with a smaller deadline, a not-urgent queue for the real-time flow with a bigger deadline, and a queue for the long flow. The extent of the urgency is determined by a prespecified deadline threshold. Simulation results show that DAQ achieves a high performance for both flows.

Although there are existing efforts, in the paper we design a flexible algorithm for data center to allocate available bandwidth with the goal of guaranteeing the real-time performance. In order to achieve this goal, we introduce a non-cooperative game-theoretic framework to schedule the data center network. The use of game theory in wireless communication, resource allocation, and social networks has been attracting extensive attentions [40,38]. In data center networks, there are also strategies that are based on game theory. In [33] a Stackelberg game is presented to deal with the problem of resource allocation with the requirements of minimizing energy consumption in data centers. In [47] Stackelberg game is used to model the problem of energy-aware resource allocation in data centers. A non-cooperative game is formulated as these decentralized scheduler compete in the resource allocation. In [22] bandwidth allocation in IaaS Datacenters is modeled as a cooperative game. In order to provide Virtual Machine (VM)-level bandwidth guarantees, asymmetric Nash bargaining solution is proposed. The main spark of our paper is to introduce game theory to model the problem of bandwidth allocation with real-time constraints. Then the paper derives a solution of bandwidth allocation by computing NE point of the game.

3. System model

In the paper, we aim to propose a Deadline-aware Rate Allocation (DRA) algorithm for scheduling heterogeneous flows in DCNs. The goals of the resource allocation and congestion control in the data centers are described as follows:

- Deadline Satisfaction: The mechanism should try to satisfy the deadline constraints for short flows as more as it could.
- Maximize Total Throughput: Besides the short flows, there are long flows that require the desired throughput performance. To allocate the bandwidth to the long flows and to maximize the total throughput is necessary.
- Dynamical Adaption: Part of the traffic is characterized by the bursty feature. Meanwhile, some servers are known as hot nodes [15]. The protocol should compromise with these factors.
- Traffic Arrival Tolerance: The performance of the protocol should be robust so that it can accommodate the flows with any arrival orders. As pointed in [24], the performance of D^3 degrades if the arrival order of flows changes.
- Low complexity: The implementation of the protocol should be easy with low cost. The algorithm with low complexity is preferred.

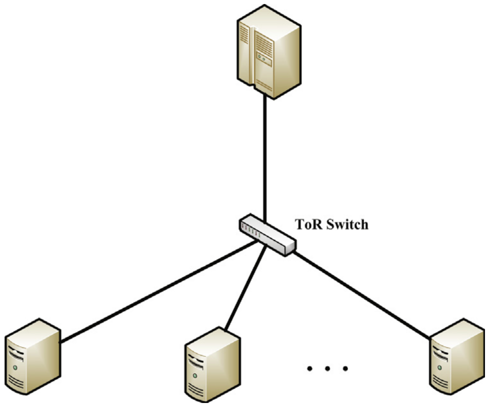


Fig. 2. A simple DCN topology.

DRA aims to minimize the amount of deadline-missing flows for short traffic. It satisfies the above mentioned features.

3.1. Network model

We hope the algorithm is independent on the implementation of a DCN. The many-to-one traffic pattern, which is common in many important data-center applications, is the basic transmission model. We divide the flows in DCNs into two categories to simplify the analysis in the following sections: the real-time flow with deadline, and the non-real-time flow without deadline. This is a common-used way to classify the traffic in data centers [45,17,15].

We consider the DCN in a discrete-time system. The DCN runs in discrete time with time slot $t \in \{0, 1, 2, \ldots\}$. The whole DCN consists of N workers and 1 server. They are connected via a switch, such as the Top-of-Rack (ToR). Fig. 2 shows a DCN example. Let $\mathcal{V}$ is the set of workers, and $\mathcal{L}$ is the set of links from the workers to the switch. Each link L_i, $i \in \mathcal{L}$, represents a transmission-pair between worker i and the switch. Meanwhile, there is a link from the switch to the server, denoted as L_0. All the links, including L_i, $i \in \mathcal{L}$, and L_0, are capacity-constrained. In general, the capacity of link L_0 and the buffer size of the switch are the bottlenecks of the communication. Each link L_i, $i \in \mathcal{L}$, maintains a queue for data convergence. When a flow is generated by worker i, it is placed into the queue for link L_i. For ease of exposition, in the following sections the link and its maintaining queue are not distinguished. The two notations are interchangeably used.

Every flow F_{ij} is constrained by the following parameter: D_{ij}, S_{ij}. Here j represents the flow number in the link i. D_{ij} is the deadline of the flow. For the real-time flow the value is assigned by SLAs. For the non-real-time one, it is set to a default value. If the real-time flow is not successfully received before its deadline, it is discarded. S_{ij} is the size of the flow. We represent the size of flow in basic resource unit that is allocated. Specially, let S'_{i1} denote the remaining size of HOL flow in link i.

Let $Q_i(t)$ be the queue backlog of links i in terms of flows at the beginning of slot t. $Q(t) = (Q_1(t), Q_2(t), \ldots.)$ represents the queue backlog vector. $Q_i(t)$ is a non-negative integer and $Q_i(0) = 0$. As we focus on the rate allocation issue among links, it is assumed each link has an infinite buffer. The flow in each worker is generated by IoT service queries or data update. Two assumptions are made about the flow-arriving process. First, the flow arrives at the end of each slot. This means that the new flow cannot be transmitted during that slot. Second, the flow can be partitioned into multiple parts and transmitted one by one. During each slot, all the workers contend to transmit. Let $S(t) = (S_1(t), S_2(t), \ldots.)$ be the served vector during slot t. When $S_i(t)$ takes the value 1, L_i is served. Let $R_i(t)$ represent the allocated bandwidth to link i during slot t. $R_i(t)$ determines the upper boundary of bandwidth to link i. At the end of each slot t, L_i drops the real-time flows that are missing deadlines at t. The vector is represented as $D(t) = (D_1(t), D_2(t), \ldots.)$. $D_i(t)$ is a non-negative integer and measures the amount of dropped flows. It is noted that if part of a real-time flow has been successfully transmitted while the remaining part expires, the flow is considered missing its deadline.

In the paper we focus on the issue of bandwidth allocation. We assume that the sender can receive ACK message from the receiver once the transmission is over. Efficient congestion control strategies are taken to modify the size of the congestion window.

3.2. Metrics

We introduce the performance metrics to evaluate the algorithms in this subsection. Application throughput and fairness index are key metrics to evaluate the algorithms. Application throughput describes the number of flows that are successfully transmitted before their deadlines. The metric is widely used to evaluate the protocol performance when the real-time flows exist [45,24,17]. We first define the deadline missing ratio of each link.

Definition 1. Deadline missing ratio R_i^{dm} of link i is defined as follows:

$$R_i^{dm} = \lim_{t\to+\infty} \frac{\sum_{k=1}^{t} D_i(k)}{\sum_{k=1}^{t} A_i(k)}. \tag{1}$$

The metric describes the long-term flow dropping ratio of each link due to the flow expiration. Based on the deadline missing ratio, the application throughput can be defined as follows.

Definition 2. Application throughput is the percentage of flows that are successfully received before their deadlines,

$$T_{ap} = \lim_{t\to+\infty} \frac{\sum_{i\in I_{rt}} \sum_{k=1}^{t} A_i(k) - \sum_{i\in I_{rt}} \sum_{k=1}^{t} D_i(k)}{\sum_{i\in I_{rt}} \sum_{k=1}^{t} A_i(k)} \tag{2}$$

$$= 1 - \lim_{t\to+\infty} \frac{\sum_{i\in I_{rt}} \sum_{k=1}^{t} D_i(k)}{\sum_{i\in I_{rt}} \sum_{k=1}^{t} A_i(k)} \tag{3}$$

where I_{rt} represents the set of real-time flows. Deadline missing ratio evaluates the performance of single link. Meanwhile, application throughput evaluates the system-level performance.

Fairness index is used to evaluate the fairness of a specified algorithm when the resource is shared among multiple competitors [26]. The metric is computed based on throughput.

Definition 3. Throughput T_i of link i is defined as follows:

$$T_i = \lim_{t\to+\infty} \sum_{k=1}^{t} S_i(k) R_i(k). \tag{4}$$

Then fairness index can be defined as follows.

Definition 4. Fairness index F^{in} of a system is defined as follows [26]:

$$F^{in} = \frac{(\sum_{i\in\mathcal{E}} T_i)^2}{E * \sum_{i\in\mathcal{E}} T_i^2} \tag{5}$$

where $\mathcal{E}$ is the set of total links that contribute to fairness index. E represents the cardinality of set $\mathcal{E}$.

In order to evaluate the time that flows eclipse in the queues, average waiting time is introduced.

Definition 5. For link i, the average waiting time of successful transmissions is

$$W_i = \frac{1}{|L_i(s)|} \sum_{j\in L_i(s)} w_{ij} \tag{6}$$

where w_{ij} is the waiting time of flow F_{ij} since it is generated. $L_i(s)$ is the flow set that flow i has successfully transmitted. $|L_i(s)|$ denotes the cardinality of the set $L_i(s)$.

4. Non-cooperative game based deadline-aware rate allocation (DRA) strategy

In the following we present a game-theoretic pricing scheme. In the mode we assume that the ToR switch charges the link per transmission. The link evaluates the value per unit bandwidth for current slot based on traffic characteristics and link statistics. Then it proposes a price p_i per unit bandwidth. According to the value, ToR switch charges a differentiated pricing scheme per unit bandwidth. The fee depends on the traffic characteristics (i.e. the timing features) and the evaluated value. It is assumed that link i earns the utility of $U(p_i)$ from the evaluated price of p_i, where $U(p_i)$ is an increasing and concave function of p_i. Let $C(p_i)$ denote the cost that link i should pay per unit of bandwidth. Therefore, the link's payoff u_i is

$$u_i(p_i) = U(p_i) - C(p_i). \tag{7}$$

ToR switch allocates the limited bandwidth to maximize the total DCN gains. In order to optimize its own payoff, the link should evaluate the bandwidth reasonably. All the links make decisions spontaneously and independently. Their objectives are conflicting, however, their decisions are interactive. Therefore, we introduce the game-theoretic analysis to model the pricing procedure. Game theory stems from economics [19]. In the last years game theory has been successfully applied in the fields of resource allocation. Researchers from computer science and electrical engineering have been modeling and analyzing the problems in the resource-limited systems with the aid of game theory. In the paper the game-theoretic model can be defined as follows:

Definition 6. A game-theoretic deadline-aware rate allocation model $\mathcal{G}$ is a L-player non-cooperative static game with the following properties:

- Players: There are L players in the game. Each player $i \in \mathcal{L}$ represents a link.
- Type space $\mathcal{D}$: D_i is an integer and takes value from $[0, D_{max}]$. It represents the remaining lifetime of the HOL flow. D_{max} is the upper bound of flow deadline.
- Strategy space $\mathcal{P}$: Player i's strategy p_i is a function of its type. It is the unit bandwidth value that each link evaluates. We assume that $p_i \geq 0$. $p_i = 0$ only happens when the backlog of the link is empty.
- Payoff space $\mathcal{U}$: Player i's payoff is $u_{p_i} = U_{p_i} - C_{p_i}$.

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.

The payoff of link i is the difference between the utility and the cost of transmission. ToR switch also considers the historical statistics of each link when it charges the link. In the paper the historical-missing ratio $M_i^K(t)$ of link i is introduced to measure the information [37]. The metric is the ratio of the total amount of flows that missing deadlines in the last K HOL flows at slot t. It is described as follows:

$$M_i^K(t) = \frac{\sum_{j=1}^{K} I_{ij}(w_{ij} \geq D_{ij})}{K} \tag{8}$$

where $I_{ij}(\cdot), j \in K$, is indicator function, satisfying the following expression:

$$I(A) = \begin{cases} 1 & \text{if A is true} \\ 0 & \text{if A is false.} \end{cases} \tag{9}$$

For the real-time flow, it is discarded if $D_{ij}(t)$ decreases to zero. $M_i^K(t)$ for the real-time link i increases by one. D_{ij} is assigned a default value if the flow is non-real-time. If $D_{ij}(t)$ decreases to zero, $M_i^K(t)$ increases by one. However, the flow is not discarded until all part of the flow is successfully transmitted.

Based on the above definitions, the cost function is defined as follows:

$$C_{p_i} = \frac{p_i}{\delta_i^{D_i} + \rho_i * M_i^K} \tag{10}$$

where δ_i is the discount factor of link i. The use of discount factor is motivated by the fact that the flow with smaller D_i should be charged less. Therefore, it can be transmitted with a higher priority. ρ_i is the weight coefficient for the tradeoff between the timing property of the flow and the historical statistics of link i.

The utility function is defined as follows:

$$U_{p_i} = lnp_i. \tag{11}$$

If the link increases its evaluated price, the transmission utility increases. On the other hand, with a fixed type D_i and $M_i^K(t)$, a higher evaluated price leads to a higher transmission cost. As a result, the link has to pay more for transmission. Therefore, the rate allocation strategy is needed to formulate at the link to maximize its own payoff. As the strategy selected by one link is conflicted by the strategy selected be another link, the optimization problem from the perspective of individual link can be formulated as follows:

$$Max \quad u_i(p_i, p_{-i}), \quad \forall i \in \mathcal{L} \tag{12}$$

where p_{-i} is the vector of the evaluated price for all links except link i, i.e. $p_{-i} = (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots)$.

The objective of the game is to find the equilibrium point that none of the links has the incentive to deviate. The following subsection investigates the equilibrium analysis.

4.1. Equilibrium analysis

Definition 7. A stationary strategy of link i is a mapping: $D_i \rightarrow p_i$. For the whole DCN, a stationary strategy is a mapping from a L-dimension vector $\mathbf{T} = (D_1, D_2, \ldots, D_L)$ to a L-dimension vector $p = (p_1, p_2, \ldots, p_L)$, where the ith element of p is the link i's value when the HOL type of link i is D_i.

Definition 8. Link i's Best Response (BR) is defined as follows:

$$p_i^* = argmax \quad [U_i(p_i(D_i)) - C_i(p_i(D_i))]. \tag{13}$$

From the above definition it can be concluded that link i's BR leads to the maximal payoff according to its type. It is important to guide each link to play its BR, in which case DCN achieves the Nash equilibrium of the game.

Definition 9. A strategy profile $p^* = (p_1^*, p_2^*, \ldots, p_L^*)$ is a Nash Equilibrium if no link can profit by unilaterally deviating its value from its BR strategy p_i^*, assuming every other link follows its equilibrium strategy.

If equilibrium state is satisfied, it can derive the following inequality:

$$\begin{aligned} &E_{D_i}[U_i(p_i^*(D_i, D_{-i})) - C_i(p_i^*(D_i, D_{-i}))] \geqslant \\ &\quad E_{D_i}[U_i(p_i(D_i, D_{-i})) - C_i(p_i(D_i, D_{-i}))] \end{aligned} \tag{14}$$

where $E_x[y]$ denotes y's expectation under the type x, D_{-i} is the type vector for all links except link i, i.e. $D_{-i} = (D_1, \ldots, D_{i-1}, D_{i+1}, \ldots)$. The constraint expression (14) forces each rational self-interested link to choose its best response p_i^*. The optimal mechanism derives u_i to its maximum value [19].

Theorem 1. *Nash Equilibrium (NE) exists in the game.*

Proof. According to the definition of the game model, u_i is described as follows:

$$u_i = lnp_i - \frac{p_i}{\delta_i^{D_i} + \rho_i * M_i^K}. \tag{15}$$

It is obvious that strategy space $\mathcal{P}$ is a nonempty, convex and compact subset in the Euclidean space $\mathcal{R}^L$. Then the second-order partial derivative of u_i with respect to p_i satisfies the following expression:

$$\frac{\partial^2 u_i}{\partial p_i^2} = -\frac{1}{p_i^2}. \tag{16}$$

As described in the game definition, $p_i = 0$ corresponds to the fact that the transmitting link of the worker is empty. The work has no flow to transmit. In the case the link quits the game. Therefore, $\frac{\partial^2 u_i}{\partial p_i^2} < 0$. u_i is a continuous and quasi-concave function in $\mathcal{P}$. According to [36,44], NE exists in the game. □

Theorem 2. *The game has unique equilibrium.*

Proof. Theorem 1 has proved that NE exists in the game. In game theory, the best response p_i^* of player is solved by the first-order partial derivative of u_i with respect to p_i, which can be represented as follows:

$$\frac{\partial u_i}{\partial p_i} = \frac{1}{p_i} - \frac{1}{\delta_i^{T_i} + \rho_i * M_i^K}. \tag{17}$$

By assuming (17) equal to zero,

$$\frac{\partial u_i}{\partial p_i} = 0. \tag{18}$$

The best response of each link can be derived. It can be found that there is only one solution in (18). Therefore, it can be concluded that the game has a unique equilibrium. □

When link i plays its best response, p_i^* is

$$p_i^* = \delta_i^{D_i} + \rho_i * M_i^K. \tag{19}$$

From the solution (19) of best response we find that p_i is the function of D_i, δ_i, ρ_i, and M_i^K. If we consider the workers are equal, no priorities exist among the workers. δ_i and ρ_i take the same values for all the links. Then the game participants can derive symmetric NE strategies. The symmetric feature in turn imposes all the links to play their best responses concurrently in the game. Therefore, all the links achieve maximal profit if they follow their best responses. In the case, the strategy can achieve NE of the game [4].

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.

4.2. Sorting strategy within queues

The workers sort the flows within their own links first. Algorithm 1 is the sorting algorithm within a link. At the first step, evaluated price of the flow to be inserted is calculated. The value is compared with that of HOL flow. If the price is not bigger, then the new flow is compared with the next flow. This process is repeated until the new flow is inserted to the position between the two flows—after the flow with a bigger or equal price and prior to the flow with a smaller price. The sorting strategy within queues makes sure that the flow is assigned a priority according to its price. $M_i^K(t)$ is the same in single link. p_{ij} is determined by δ_i and $D_{ij}(t)$. In the game we assume the symmetric feature exist. In the case the flows with different deadlines in L_i and different query responses take the same value of discount factor. Then the ordering algorithm within a link equals to the conventional EDF scheme.

Algorithm 1 Biggest Price First Algorithm

```
Input:
    Queue $L_i$ that has been sorted before slot $t$;
    New arriving flow, $F_{ij}$;
Output:
    Queue $L_i$ with new flow inserted;
$k = 1, n = Q_i(t)$, computing $p_{ij}$;
point to the HOL flow of $L_i$;
while $k \le n$ and $p_{ij} \le p_{ik}$
    do    $k{+}{+}$;
if $p_{ij} > p_{ik}$
    insert $F_{ij}$ into queue $L_i$ prior to $F_{ik}$
else if $k > n$
    insert $F_{ij}$ into queue $L_i$ after $F_{ik}$
return $L_i$
```

4.3. Bandwidth allocation strategy

The purpose of scheduling policy in DCNs is to satisfy the timing constraints of the IoT service. Meanwhile, it should maximize system throughput. The aim of bandwidth allocation is to find out the solution of the following optimization problem:

$$u = \sum_{i \in \mathcal{L}} u_i S_i R_i \tag{20}$$

where u is the summation of u_i in the current slot. It is bounded by the bandwidth capacity in the datacenter. R_i is the allocated bandwidth to link i. To avoid wasting bandwidth, the value of allocated bandwidth should be less than or equal to the remaining size of HOL flow, i.e.

$$R_i \le S'_{i1}. \tag{21}$$

Our objective is to achieve the upper bound of u by allocating bandwidth at each slot. We can formulate the problem of optimization performance while we can guarantee the performance-centric fairness as follows:

$$Max \quad u \tag{22}$$

$$s.t.$$

$$S_i = \{0, 1\} \quad \forall i \tag{23}$$

$$\sum_{i \in \mathcal{L}} R_i \le C \tag{24}$$

$$R_i \le S'_{i1} \quad \forall i \in \mathcal{L} \tag{25}$$

$$R_i \le C_i \quad \forall i \in \mathcal{L}. \tag{26}$$

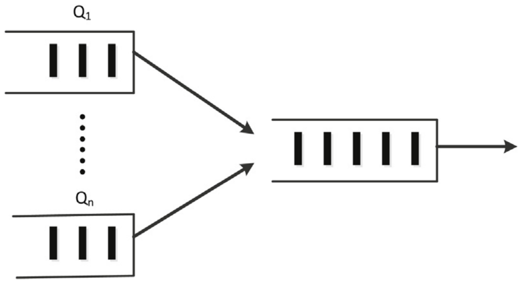


Fig. 3. Link structure in DRA.

The first constraint ensures that each link is either served or not. The second constraint ensures that the summation of the allocated bandwidth does not exceed the upper bound of the bandwidth capacity. The third constraint ensures the allocated bandwidth is not wasted. The last constraint ensures that the allocated bandwidth does not exceed the specific link capacity.

In order to solve the (22)–(26), DRA is proposed in the paper. DRA works in a greedy manner as follows, which can be seen in Fig. 3. (1) When each work is ready to transmit, it sends a SYN packet for the initialization of a transmission. The information of the link, i.e. the evaluated price and the remaining size of HOL flow, is attached to the header of SYN packet. (2) ToR switch detaches the header of the SYN packet. Then it orders the links based on evaluated price in descending order. ToR switch checks the link list in sequence and assigns the desired bandwidth to the link if the residual bandwidth is adequate. If the residual bandwidth is inadequate, the switch assigns all the residual bandwidth to the link. For each flow, ToR switch responds a SYN/ACK packet to the link with the information of allocated bandwidth. (3) The link uses the information to determine its sending rate that transmits the HOL flow. (4) When the slot ends, all the links go back to step 1 to transmit the remaining flows or new flows. The whole procedure of the algorithm for bandwidth allocation is shown in Algorithm 2. When there is a tie that two or more flows have the same price, tie-breaking is done according to link ID. The link with a smaller ID wins the tie.

Algorithm 2 Deadline-aware Rate Allocation (DRA) Algorithm

```
Input:
    Total available capacity C, information of HOL flow and link;
Output:
    Rate allocated vector to each flow;
1:  Switch acquires the price vector and then sort the vector by
    price in descending order.
2:  point to the first element in price vector;
3:    If $C \ge S'_{i1}$, then
4:      if $S'_{i1} \le C_i$
5:        $R_i = S'_{i1}, C = C - R_i$;
6:      else
7:        $R_i = C_i, C = C - C_i$;
8:      point to the next element, go to 3
9:    else
10:     if $C > C_i$,
11:       $R_i = C_i, C = C - C_i$, point to the next element, go to 3
12:     else
13:       $R_i = C_i, C = 0$;
14: return allocated rate vector
```

Without loss of generality, we make the following three assumptions: (1) there is no packet loss. (2) Each link can only transmit the HOL flow at a slot. (3) All the links from the workers to the switch have the same capacity.

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.

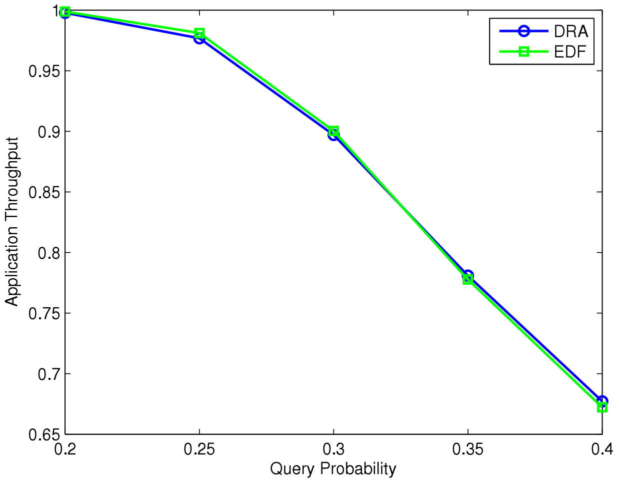


Fig. 4. Impact of query probability on application throughput.

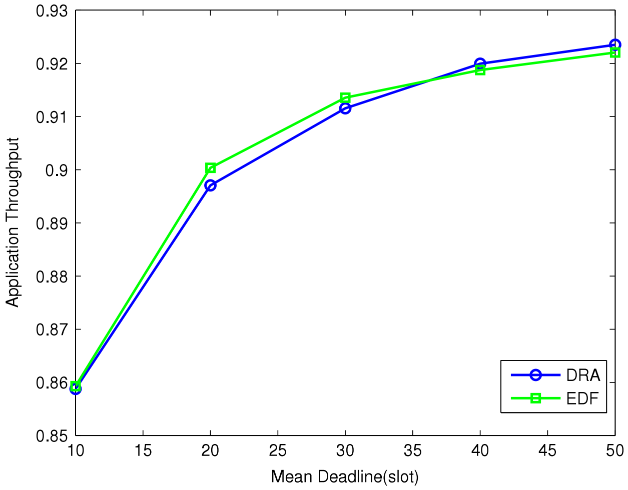


Fig. 5. Impact of flow deadline on application throughput.

5. Performance evaluation

In this section, we evaluate the performance of the DRA algorithm and compare it with the ideal algorithm, i.e. EDF, in DCNs. EDF is shown to be the optimal scheduling algorithm for sporadic tasks constrained by deadlines [16]. As the paper focuses on real-time performance, EDF is a natural choice in the performance comparison.

5.1. Simulation setup

In the simulations, a two-tie tree-topology scenario is considered. The scenario includes multiple workers. These workers connect to an aggregator via a switch. We assume that most 100 packets can be transmitted per time-slot. Meanwhile, all the links from the workers to the switch have the same capacity. The capacity is equal to the link capacity from the switch to the aggregator. Therefore, the link from the switch to the aggregator is the performance bottleneck in DCNs. δ_i is 0.95 for all links and ρ_i takes 0.5. When simulation begins, $M_i^K(0)$ equals to 0. The parameter for historical statistics K is 10. The total simulation slot is set to 10^6.

5.2. Performance analysis

In this subsection, we analyze the real-time performance of DRA. One of the most common scenarios in DCNs is query-response, i.e. all the workers initiate flows at the same time due to the query from the aggregator. Although the flows are generated simultaneously, the traffic patterns of flows, such as flow size and deadline, are heterogeneous. The compared algorithm is the EDF algorithm. EDF allocates the bandwidth based on the flow size in the ascending order of deadline. In the paper we focus on the real-time performance. In the following simulations, the number of involved sending workers is fixed at 14.

Impact of Query Probability. We first investigate the impact of query probability on the performance of DCNs. We change the query probability to analyze its effects on the application throughput. The results can be seen in Fig. 4. More queries lead to higher amount of traffic. When the total bandwidth keeps constant, the application throughput is inversely proportional to the query probability. In the comparisons the flows are random generated. We take moderate deadlines, i.e. deadline takes value from 1 slot to 40 slots. The flow size is uniformly distributed from 1 packet to 50 packets.

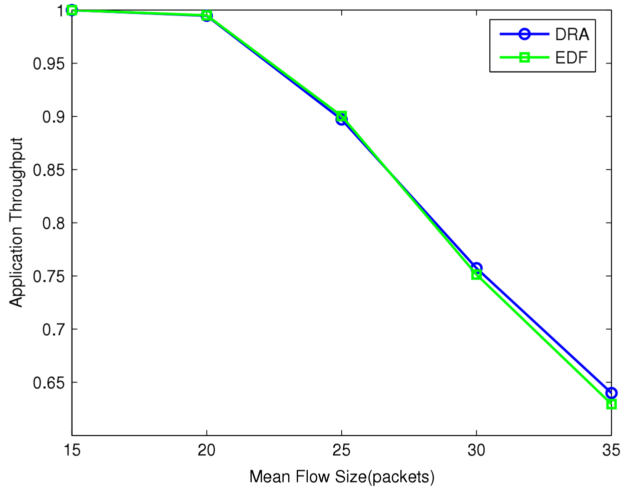


Fig. 6. Impact of flow size on application throughput.

Impact of Deadline. We further investigate the impacts of traffic deadline. We vary the deadlines with mean value between 10 slots to 50 slots. The query probability is fixed at 0.3 and the number of workers is 14. Fig. 5 shows the result. A long deadline means the flow has long survival time, which increases its transmission chance. Therefore, the application throughput is proportional to the mean value of the deadline.

Impact of Flow size. We further investigate the impacts of flow size. In DRA when the evaluated price vector is determined, the switch allocates the available bandwidth according to the flow size. We vary the flow size with mean between 15 packets and 35 packets. The query probability is fixed at 0.3. The deadline takes value from 1 slot to 40 slots. Its mean value is 20 slots. The number of workers is 14. Fig. 6 shows the result.

5.3. Discussion

From the above analysis it can been concluded that the DRA almost has the same application throughput with EDF. In order to demonstrate the real-time performance of DRA, we further investigate the two algorithms.

Fairness. Figs. 7–9 show the fairness index. In order to prevent the link starvation and provide the fair allocation, we use the evaluated

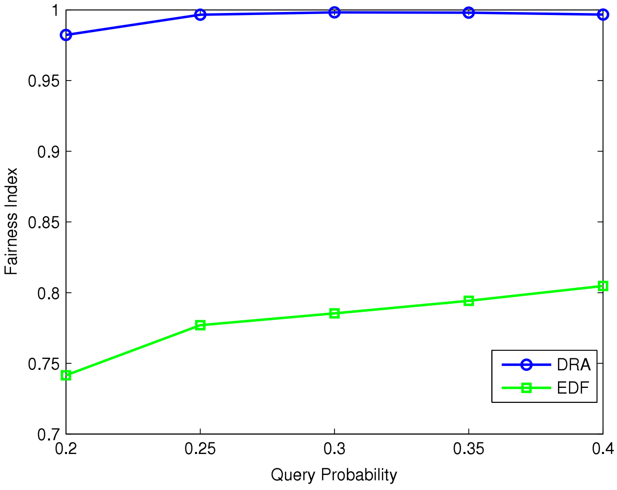


Fig. 7. Fairness index with varying query probabilities.

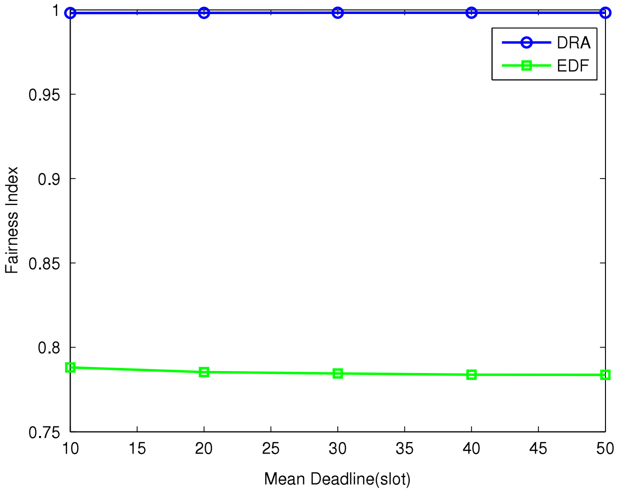


Fig. 8. Fairness index with varying flow deadlines.

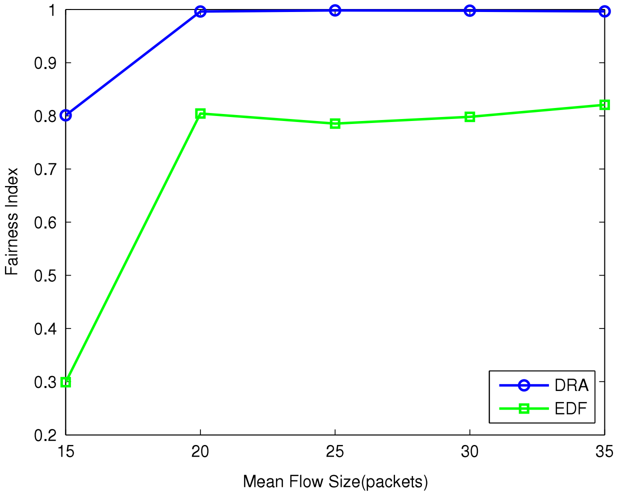


Fig. 9. Fairness index with varying flow sizes.

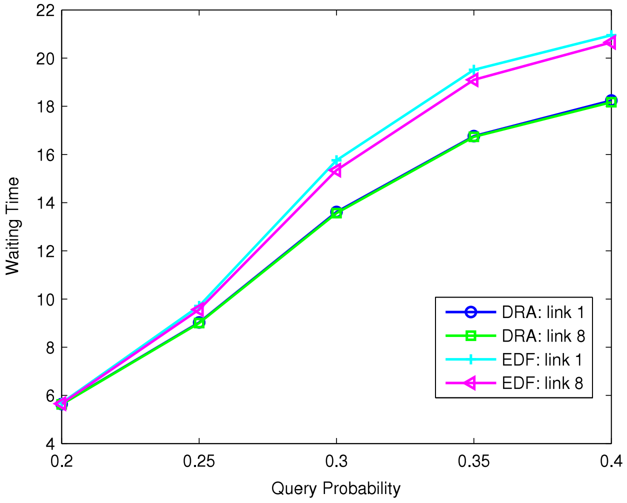


Fig. 10. Average waiting time with varying query probabilities.

price to allocate the bandwidth. The parameter takes the timing property of the flow and the historical statistics of the link into consideration. When a link is allocated less bandwidth in the past period, DRA provides the mechanism to guarantee that the priority of the link increases. Therefore, the link has the opportunity to be allocated bandwidth with higher priority. From the perspective of fairness performance, DRA is obviously superior to EDF.

Average waiting time. Then we evaluate the average waiting time of a flow, i.e. the period between the time a flow arrives at a link and the time it is successfully transmitted. We still set the default value of query probability at 0.3 and the default value of mean deadline at 20 slots. The flow size is uniformly distributed from 1 packet to 50 packets in Figs. 10–11. The number of workers is 14. In the tree-like topology, all the links are equal and have the same priority of bandwidth allocation from the perspective of network topology. DRA assigns the priorities during the running-stage. Therefore, we can randomly select the comparing links to show the results. The experimental results, that link 1 and link 8 achieve similar performance, demonstrate the point. From Fig. 10 to 12 it can be concluded that DRA achieves smaller waiting time in all the scenarios. Although there is quite a small difference in the respective application throughput that DRA and EDF achieves, the real-time performance of DRA is better than that of EDF.

6. Conclusion

In order to support various IoT services to users, it requires the data center to provide rich and efficient computing. Due to the flow characteristics in DCNs, there are several challenges for the data center networking. First, the visiting from end-user is non-deterministic. Second, the flows require heterogeneous QoS goals. The applications in data centers contribute to time-sensitive flows and throughput-sensitive flows. In the paper we aim to design a policy that can guarantee the real-time performance for flows. Based on non-cooperative game-theoretic analysis, the DRA algorithm for scheduling the heterogeneous flows in DCNs is proposed. The algorithm makes a tradeoff between the timing parameters and the historical statistics of each link when the bandwidth is allocated. The simulation results have shown that the mechanism guarantees the real-time performance while it achieves good fairness and latency performance.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (61572466) and in part by the Beijing Natural Science Foundation (4162059).

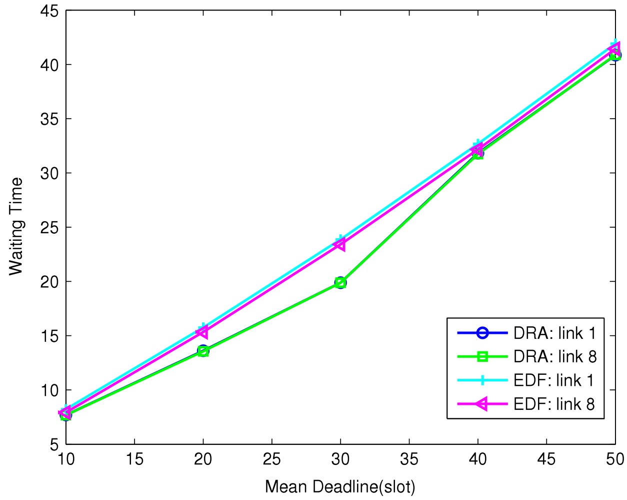


Fig. 11. Average waiting time with varying flow deadlines.

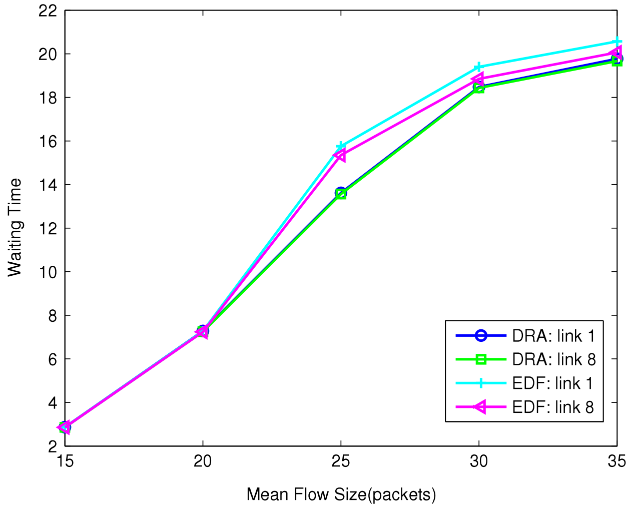


Fig. 12. Average waiting time with varying flow sizes.

References

[1] M. Al-Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, in: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, SIGCOMM '08, ACM, New York, NY, USA, 2008, pp. 63–74.

[2] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, M. Sridharan, Data center tcp (dctcp), in: Proceedings of the ACM SIGCOMM 2010 Conference, SIGCOMM '10, ACM, New York, NY, USA, 2010, pp. 63–74.

[3] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, S. Shenker, pFabric: Minimal near-optimal datacenter transport, in: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13, ACM, New York, NY, USA, 2013, pp. 435–446.

[4] A. Antonopoulos, C. Verikoukis, Multi-Player game theoretic mac strategies for energy efficient data dissemination, IEEE Trans. Wireless Commun. 13 (2) (2014) 592–603.

[5] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, et al., A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.

[6] M.D. Assuncao, R.N. Calheiros, S. Bianchi, M.A. Netto, R. Buyya, Big data computing and clouds: Trends and future directions, J. Parallel Distrib. Comput. 79–80 (2015) 3–15.

[7] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805.

[8] H. Ballani, P. Costa, T. Karagiannis, A. Rowstron, Towards predictable datacenter networks, in: Proceedings of the ACM SIGCOMM 2011 Conference, SIGCOMM '11, ACM, New York, NY, USA, 2011, pp. 242–253.

[9] J.M. Batalla, M. Gajewski, W. Latoszek, P. Krawiec, C.X. Mavromoustakis, G. Mastorakis, ID-based service-oriented communications for unified access to IoT, Comput. Electr. Eng. 52 (2016) 98–113.

[10] T. Chen, X. Gao, G. Chen, The features, hardware, and architectures of data center networks: A survey, J. Parallel Distrib. Comput. 96 (2016) 45–74.

[11] B.-W. Chen, X. He, W. Ji, S. Rho, S.-Y. Kung, Support vector analysis of large-scale data based on kernels with iteratively increasing order, J. Supercomput. 72 (9) (2016) 3297–3311.

[12] B.-W. Chen, W. Ji, S. Rho, Geo-conquesting based on graph analysis for crowd-sourced metatrails from mobile sensing, IEEE Commun. Mag. 55 (1) (2017) 92–97.

[13] B.-W. Chen, S. Rho, M. imran, M. Guizani, W.K. Fan, Cognitive sensors based on ridge phase-smoothing localization and multiregional histograms of oriented gradients, IEEE Trans. Emerging Top. Comput. (2016).

[14] B.-W. Chen, S. Rho, L.T. Yang, Y. Gu, Privacy-preserved big data analysis based on asymmetric imputation kernels and multiside similarities, Future Gener. Comput. Syst. (2016).

[15] Y. Cui, H. Wang, X. Cheng, D. Li, A. Yla-Jaaski, Dynamic scheduling for wireless data center networks, IEEE Trans. Parallel Distrib. Syst. 24 (12) (2013) 2365–2374.

[16] R.I. Davis, A. Burns, A survey of hard real-time scheduling for multiprocessor systems, ACM Comput. Surv. 43 (4) (2011) 35:1–35:44.

[17] C. Ding, R. Rojas-Cessa, DAQ: Deadline-aware queue scheme for scheduling service flows in data centers, in: ICC, 2014, pp. 2989–2994.

[18] E. Feller, L. Ramakrishnan, C. Morin, Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study, J. Parallel Distrib. Comput. 79–80 (2015) 80–89.

[19] D. Fudenberg, J. Tirole, Game Theory, MIT Press, 1991.

[20] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, S. Sengupta, VL2: A scalable and flexible data center network, in: Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, SIGCOMM '09, ACM, New York, NY, USA, 2009, pp. 51–62.

[21] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.

[22] J. Guo, F. Liu, J.C.S. Lui, H. Jin, Fair network bandwidth allocation in iaas datacenters via a cooperative game approach, IEEE/ACM Trans. Netw. 24 (2) (2016) 873–886.

[23] B. Hayes, Cloud computing, Commun. ACM 51 (7) (2008) 9–11.

[24] C.-Y. Hong, M. Caesar, P.B. Godfrey, Finishing flows quickly with preemptive scheduling, SIGCOMM Comput. Commun. Rev. 42 (4) (2012) 127–138.

[25] J. Hwang, J. Yoo, N. Choi, Deadline and incast aware TCP for cloud data center networks, Comput. Netw. 68 (2014) 20–34.

[26] R. Jain, D.-M. Chiu, W.R. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System, Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

[27] W. Ji, Y. Chen, M. Chen, B.W. Chen, Y. Chen, S.Y. Kung, Profit maximization through online advertising scheduling for a wireless video broadcast network, IEEE Trans. Mob. Comput. 15 (8) (2016) 2064–2079.

[28] W. Ji, B.W. Chen, Y. Chen, S.Y. Kung, Profit improvement in wireless video broadcasting system: a marginal principle approach, IEEE Trans. Mob. Comput. 14 (8) (2015) 1659–1671.

[29] W. Ji, P. Frossard, B.W. Chen, Y. Chen, Profit optimization for wireless video broadcasting systems based on polymatroidal analysis, IEEE Trans. Multimedia 17 (12) (2015) 2310–2327.

[30] W. Ji, Z. Li, Y. Chen, Joint source-channel coding and optimization for layered video broadcasting to heterogeneous devices, IEEE Trans. Multimedia 14 (2) (2012) 443–455.

[31] K. Kambatla, G. Kollias, V. Kumar, A. Grama, Trends in big data analytics, J. Parallel Distrib. Comput. 74 (7) (2014) 2561–2573.

[32] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, R. Chaiken, The nature of data center traffic: Measurements & analysis, in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, IMC '09, ACM, New York, NY, USA, 2009, pp. 202–208.

[33] X. Leon, L. Navarro, A Stackelberg game to derive the limits of energy savings for the allocation of data center resources, Future Gener. Comput. Syst. 29 (1) (2013) 74–83.

[34] J. Mongay Batalla, C.X. Mavromoustakis, G. Mastorakis, K. Sienkiewicz, On the track of 5g radio access network for IoT wireless spectrum sharing in device positioning applications, in: C.X. Mavromoustakis, G. Mastorakis, J.M. Batalla (Eds.), Internet of Things (IoT) in 5G Mobile Technologies, Springer International Publishing, Cham, 2016, pp. 25–35.

[35] A. Munir, I. Qazi, Z. Uzmi, A. Mushtaq, S. Ismail, M. Iqbal, B. Khan, Minimizing flow completion times in data centers, in: INFOCOM, 2013, pp. 2157–2165, ISSN: 0743-166X.

[36] C. Saraydar, N.B. Mandayam, D. Goodman, Efficient power control via pricing in wireless data networks, IEEE Trans. Commun. 50 (2) (2002) 291–303.

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.

[37] B. Shen, S. Rho, X. Zhou, R. Wang, A delay-aware schedule method for distributed information fusion with elastic and inelastic traffic, Inf. Fusion 36 (2017) 68–79.

[38] B. Shen, X. Zhou, M. Kim, Mixed scheduling with heterogeneous delay constraints in cyber-physical systems, Future Gener. Comput. Syst. 61 (2016) 108–117.

[39] B. Vamanan, J. Hasan, T. Vijaykumar, Deadline-aware datacenter tcp (d2tcp), SIGCOMM Comput. Commun. Rev. 42 (4) (2012) 115–126.

[40] R. Wang, W. Cai, A sequential game-theoretic study of the retweeting behavior in Sina Weibo, J. Supercomput. 71 (9) (2015) 3301–3319.

[41] G. Wang, T. Ng, The impact of virtualization on network performance of amazon ec2 data center, in: INFOCOM, 2010, pp. 1–9, ISSN: 0743-166X.

[42] R. Wang, S. Rho, W. Cai, High-performance social networking: microblog community detection based on efficient interactive characteristic clustering, Cluster Comput. 20 (2) (2017) 1209–1221.

[43] R. Wang, S. Rho, B.-W. Chen, W. Cai, Modeling of large-scale social network services based on mechanisms of information diffusion: Sina weibo as a case study, Future Gener. Comput. Syst. 74 (2017) 291–301.

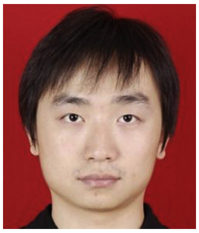
[44] F. Wang, O. Younis, M. Krunz, Throughput-oriented MAC for mobile ad hoc networks: A game-theoretic approach, Ad Hoc Networks 7 (1) (2009) 98–117.

[45] C. Wilson, H. Ballani, T. Karagiannis, A. Rowtron, Better never than late: meeting deadlines in datacenter networks, in: Proceedings of the ACM SIGCOMM 2011 Conference, SIGCOMM '11, ACM, New York, NY, USA, 2011, pp. 50–61.

[46] H. Wu, Z. Feng, C. Guo, Y. Zhang, ICTCP: Incast congestion control for TCP in data-center networks, IEEE/ACM Trans. Netw. 21 (2) (2013) 345–358.

[47] B. Yang, Z. Li, S. Chen, T. Wang, K. Li, Stackelberg Game Approach for energy-aware resource allocation in data centers, IEEE Trans. Parallel Distrib. Syst. 27 (12) (2016) 3646–3658.

[48] J. Zhang, F. Ren, C. Lin, Modeling and understanding TCP incast in data center networks, in: INFOCOM, 2011, pp. 1377–1385, ISSN: 0743-166X.



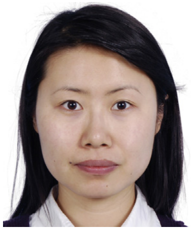
Bo Shen received his B.S. degree in Network Engineering from Xidian University, Xi'an, China, M.S. degree in Communication and Information System from Northwestern Polytechnical University, Xi'an, China, and Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China. From 2017, he is a Post Doctoral Fellow at Department of Electrical Engineering, Princeton University, NJ, USA. His current research interests include cyber–physical systems, cloud computing, wireless communication, machine learning, and big data.

Naveen Chilamkurti is currently serving as Head, Department of Computer Science and Computer Engineering in La Trobe University at Melbourne, Australia. His research areas include but are not limited to Intelligent Transport Systems (ITS), Wireless Multimedia, and Wireless Sensor Networks. He has published more than 180 Journal and conference papers. He has served as an editor for renowned International Journals including Inaugural Editor-in-Chief for International Journal of Wireless Networks and Broadband Technologies, technical editor for IEEE wireless communication magazine, associate technical editor for IEEE communication magazine, and associate editor for Wiley IJCS and SCN journals.

Ru Wang received the B.S. degree from Xidian University, Xi'an, China, the M.S. degree from the Chang'an University, Xi'an, China, and the Ph.D. degree in Computer Science from Northwestern Polytechnical University, Xi'an, China. She joined in College of Information Engineering, Northwest A&F University since 2017. Her research focuses on P2P network, social network, complex network, and large-scale data analysis.

Xingshe Zhou received the M.E. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 1983. He is currently a Professor with the School of Computer Science and Technology, North-western Polytechnical University. He is the author of more than 200 papers in his areas of interest, including embedded computing, distributed computing, and pervasive computing. He is currently an Executive Director of the China Computer Federation and a Program Committee Member of the National Natural Science Foundation of China.

Shiwei Wang received the Ph.D. degree in control systems from the Liverpool John Moores University in 2007. He is currently the R&D Director of Weihai Yuanhang Technology Development Co., Ltd., Weihai, China, as an awardee of the Thousand Talents Plan of China. His current research interests include artificial intelligent, machine vision, robotics and the corresponding application in food and pharmaceutical industry.



Wen Ji received the M.S. and Ph.D. degrees in communication and information systems from Northwestern Polytechnical University, China, in 2003 and 2006, respectively. She is an Associate Professor in the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. From 2014 to 2015, she was a visiting scholar at the Department of Electrical Engineering, Princeton University, USA. Her research areas include video communication and networking, video coding, channel coding, information theory, optimization, network economics, and ubiquitous computing.

Please cite this article in press as: B. Shen, et al., Deadline-aware rate allocation for IoT services in data center network, J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.09.012.