A Dual Price-Based Congestion Control Mechanism for Optical Burst Switching Networks

Tairan Zhang, Wei Dai, Guiling Wu, Xinwan Li, Jianping Chen, and Chunming Qiao, *Fellow, IEEE*

***Abstract*—A dual price-based congestion control (DPCC) mechanism for optical burst switching (OBS) networks is proposed in this paper, which can achieve an optimal rate-reliability tradeoff by allocating proper network traffic and resources based on the idea of network utility maximization (NUM). The DPCC uses the congestion and reliability prices and feedback information to dynamically adjust the users' data sending rate and the end-to-end data transmission reliability in an OBS network. The performance of DPCC is evaluated and analyzed through simulations. Results verify that DPCC works very well in terms of its convergence and optimality. Moreover, compared with TCP, DPCC can achieve a maximum network utility, a parameter which can be used to reflect the overall user satisfaction degree in a network. DPCC is scalable due to its distributed nature.**

***Index Terms*—Dual price-based congestion control, network utility maximization, optical burst switching.**

I. Introduction

OPTICAL burst switching (OBS) combines the advantages of optical circuit switching (OCS) and optical packet switching (OPS) while avoiding their shortcomings. It is regarded as a promising solution for future IP over WDM networks [1]. However, due to the lack of optical buffer in OBS networks, higher burst losses may occur under a normal traffic load. Many studies have been carried out to reduce the burst loss ratio (BLR) at each OBS node (edge node and/or core node) in terms of burst scheduling [2]–[5], burst assembly [6]–[8] and so on. Most of the existing approaches, however, either just offer local treatment or are reactive approaches only invoked after contention occurs. In order to improve the overall network performance further, a network-wide solution or proactive network congestion mechanism taking into consideration the overall network congestion status are needed, especially as the explosive growth of cloud computation, real-time video, online game and so on, where an extremely large number of users may send their tremendous data into the network in an out of order manner.

There exist some global congestion control mechanisms for OBS networks. In [9], an explicit feedback mechanism was proposed to reduce the BLR and increase the network utilization. However this approach cannot control the congestion sufficiently due to the unpredictable nature of traffic. Kim *et al.* [10], [11] suggested an integrated congestion control mechanism which took both proactive and reactive control into consideration, but the mechanism needs an additional flow-policing scheme in order to drop some bursts intentionally, which inevitably affects the burst loss performance. Network utility maximization (NUM) has been extensively researched as a model for distributed network rate allocation and congestion control [12]–[14]. Hence, [15], [16] tried to solve the congestion problem in OBS networks from the view of NUM. However, that scheme can merely be performed when timer-based burst assembly is used and no wavelength converter exists at each node. Moreover, the scheme only cares about the parameters related to the MAC layer (i.e., burst length and offset time) rather than those at the higher layers on the user side. As a result, too much data may be sent to the ingress node from user side in a unit time, exceeding the transmitting or processing capability of the MAC layer.

In this paper, we design a dual price-based congestion control (DPCC) mechanism for OBS networks based on the idea of NUM. Instead of adjusting the parameters at the MAC layer as in [15], [16], we directly focus on controlling the parameters related to the user side. Because the data sending rate by any user may affect the end-to-end data transmission reliability (i.e., subtract the end-to-end BLR from 1) of all users, the sending rate and reliability of the OBS network are globally coupled together across the links and users.

Accordingly, in the proposed DPCC mechanism, the user's data sending rate and the end-to-end data transmission reliability performance are adjusted together in order to maximize the network utility. A high congestion price will deter the users at the source from sending too much data into an OBS network in order to avoid or alleviate congestion. Conversely, a low congestion price will encourage the users to increase their data sending rate. Similarly, the reliability price will influence the user's willingness to pay for a certain degree of end-to-end reliability performance he expects, and in turn the data sending rates as well. If a user wants to achieve a good reliability, he should decrease his sending rate in order to reduce burst contention and the resulting dropping probability. Thus, the dual congestion and reliability prices can incentivize the users to increase or decrease their sending

Manuscript received January 5, 2014; revised May 19, 2014 and March 23, 2014; accepted February 25, 2014. Date of publication May 29, 2014; date of current version June 23, 2014. This work was supported in part by the National 973 Program of China (2011CB301700), the National Natural Science Foundation of China (61071011, 61127016), the STCSM Project (12XD1406400), and the International S&T Cooperation Program from the Ministry of Science and Technology of China (2011DFA11780).

T. Zhang, G. Wu, X. Li, and J. Chen are with the State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ztr1918294@163.com; wuguiling@sjtu.edu.cn; lixinwan@sjtu.edu.cn; jpchen62@sjtu.edu.cn).

W. Dai is with the Networked Systems at University of California, Irvine, CA 92697 USA (e-mail: daiweix@gmail.com).

C. Qiao is with the CSE Department, State University of New York, Buffalo, NY 14228 USA (e-mail: qiao@computer.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2014.2327171

rates for the purpose of performing congestion control and achieving an appropriate reliability.

Meanwhile, the OBS link side also updates related parameters such as the maximum rate that can be accepted by a output link *l* of a core node from its input port *n* in real time according to the congestion and reliability prices. In addition, these two prices will be adjusted according to the current sending rate, reliability and related parameters from the link side.

In this way, the entire OBS network consisting of users and links can be regarded as a feedback control loop system, and the proposed DPCC mechanism will eventually achieve an optimal rate-reliability tradeoff in its stable state, which maximizes the overall network utility (i.e., the overall user satisfaction degree in a network).

The rest of the paper is organized as follows. Section II formulates the NUM problem based on the OBS link model considering the streamline effect at a bufferless OBS node. In Section III, a DPCC mechanism for OBS networks is designed to solve the NUM problem by using the *Lagrangian* method [17], [18], and its control procedure and scalability are discussed here. Section IV presents the simulation results which show how an OBS network is able to obtain the optimal rate-reliability. After that, as a further discussion, the proposed DPCC is compared with TCP to show its specialties and the advantage of achieving the best utility in Section V. The last section draws the conclusions.

II. Problem Formulation

Assuming that an OBS network is composed of a set of links $\mathbf{L} = (1, 2, \ldots, l, \ldots, L)$, where each link l has several data channels. These link resources are shared by a set of users $\mathrm{S} = (1, 2, \ldots, s, \ldots, S)$, where each user s sends a traffic flow from source to destination. Every user has a utility function $U_s(x_s, R_s)$ where x_s is the data sending rate and R_s is the end-to-end reliability performance that the user s would like to have. Since the utility reflects the overall user satisfaction degree of the OBS network, the higher the utility is, the better the rate-reliability is. It is well known that a strictly concave function always has a unique maximum over a closed and bounded constraint set. Hence, utility function should be a continuous concave function of x_s and R_s [19], [20]. Since packets are likely to be lost in an OBS network due to burst collisions, R_s is constrained by the BLR on each OBS link. Thus, it can be expressed as (1) when $P_{s,l} \ll 1$

$$R_s = \prod_{l \in L(s)} (1 - P_{s,l}) \leq 1 - \sum_{l \in L(s)} P_{s,l} \tag{1}$$

where $L(s)$ is the set of all links on the path from source to destination for user s, and $P_{s,l}$ is the BLR of traffic flow from user s on link l.

In OBS networks, N flows may go into N input ports of a switching node simultaneously. Due to the fact that no optical buffer exists inside an OBS network, they will contend with each other when sharing the resource of the same output link l. After that, bursts in those flows are streamlined onto that common link l and do not contend with each other until they diverge,

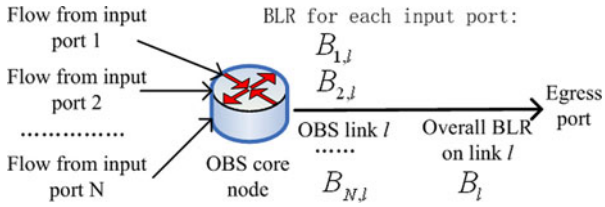

Fig. 1. The OBS link model considering streamline effect at a bufferless OBS node.

which means once the contentions among them are resolved at the first link where they merge, no intra-stream contention will occur thereafter. That phenomenon is called streamline effect and its performance at a bufferless OBS node was analyzed in [21]. Therefore, the $P_{s,l}$ should be calculated according to the streamline effect. Meanwhile, the rate allowed by each input port should be limited.

As for the user s, he can control his data sending rate x_s and ask for a reasonable reliability performance R_s according to the congestion and reliability prices. Those prices are updated and advertised dynamically as will be introduced later.

Based on the discussion above, the NUM problem can be formulated as shown in (2), where $S(n, l)$ is the set of users whose total traffic flows are going into a switching core node from input port n and out of that node to output link l (see Fig. 1), and $X_{n,l}$ is the maximum rate that can be accepted by link l from input port n. The first constraint specifies the bounds on x_s. The second constraint shows that the total traffic sent by all the users from $S(n, l)$ should not be larger than $X_{n,l}$. The last constraint means the best reliability performance that user s can obtain in an OBS network is constrained by the end-to-end BLR and the lower limit

$$\begin{aligned} &\max \sum_s U_s(x_s, R_s) \\ &\text{subject to}: 0 \leq x_{\min} \leq x_s \leq x_{\max} \\ &\qquad 0 \leq \sum_{s \in S(n,l)} x_s \leq X_{n,l} \\ &\qquad 0 \leq R_{\min} \leq R_s \leq 1 - \sum_{l \in L(s)} P_{s,l}. \end{aligned} \tag{2}$$

III. The Congestion Control Mechanism

A. Theoretical Analysis

Duality theory [17] is a good way to solve the above optimization problem like (2), but the convexity requirement in the constraints should be met at first. Similar to the analysis in [19], [20], we need to ensure $P_{s,l}$ is an increasing convex function of $X_{n,l}$. Because $\sum_{s \in S(n,l)} x_s$ will converge and finally be equal to $X_{n,l}$, it is reasonable to assume $\sum_{s \in S(n,l)} x_s = X_{n,l}$. Hence, the relation between $P_{s,l}$ and $X_{n,l}$ could be derived according to the OBS link model considering streamline effect at a bufferless OBS node. As shown in Fig. 1, the BLR at input port n can be expressed as (3) when the input burst traffic is a

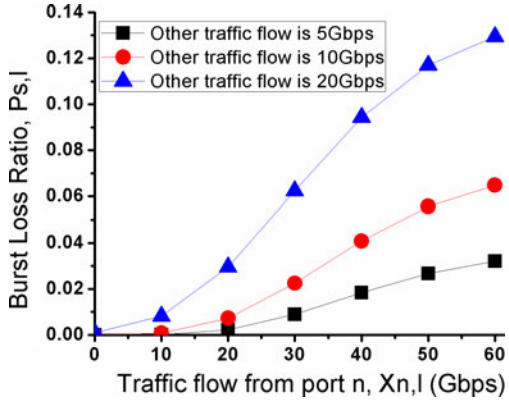

Fig. 2. The relation between $P_{s,l}$ and $X_{n,l}$

Possion flow

$$B_{n,l} = \frac{D\left(\sum_{s\in S(l)} x_s\right) - D\left(\sum_{s\in S(n,l)} x_s\right)}{1 - D\left(\sum_{s\in S(n,l)} x_s\right)} = \frac{D\left(\sum_{n'=1}^{N} X_{n',l}\right) - D(X_{n,l})}{1 - D(X_{n,l})}$$
$$D(x) = \frac{(x/W)^K \Big/ K!}{\sum_{i=0}^{K} (x/W)^i \Big/ i!} \tag{3}$$

where N is the number of input ports, K is the number of data channels in each OBS link, and W is the bandwidth of each data channel. $S(l)$ is the set of all users that share OBS link l. $B_{n,l}$ is the BLR of input n to link l. Since all the users from the same input port have the same loss ratio, it is easy to get

$$P_{s,l} = B_{n,l} = \frac{D\left(\sum_{n'=1}^{N} X_{n',l}\right) - D(X_{n,l})}{1 - D(X_{n,l})}, \quad \forall s \in S(n,l). \tag{4}$$

The relation between $P_{s,l}$ and $X_{n,l}$ has been clear, but we still cannot show that $P_{s,l}$ is an increasing convex function of $X_{n,l}$. Nevertheless, the convexity property can be satisfied when $P_{s,l}$ is relatively small (less than 6%), which fits most practical cases where the BLR is required to be much low. Detailed analysis is provided here. Assuming each wavelength bandwidth is 10 Gb/s, $\left(\sum_{n'=1}^{N} X_{n',l}\right) - X_{n,l}$ (traffic flows from other ports rather than port n) is 5, 10, and 20 Gb/s, respectively. We can plot Fig. 2 according to (4). It is demonstrated in Fig. 2 that $P_{s,l}$. is a sigmoid increasing function of $X_{n,l}$. Therefore, $P_{s,l}$. is a convex increasing function of $X_{n,l}$ when the BLR is relatively small, which also satisfies the first constraint in (2). Considering the BLR in practical OBS networks is always much low (not larger than 6%), we can conclude from Fig. 2 that the convexity property between $P_{s,l}$ and $X_{n,l}$. is usually satisfied.

Then, we adopt a dual decomposition approach to solve the optimization problem defined in (2). Specifically, the *Lagrangian* function associated with the objective function in (2) can be written as follows:

$$L(\mathbf{x},\mathbf{R},\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu}) = \sum_s U_s(x_s,R_s) + \sum_l \sum_{n\in N(l)} \lambda_{n,l}\left(X_{n,l} - \sum_{s\in S(n,l)} x_s\right) + \sum_s \mu_s\left(1 - \sum_{l\in L(s)} P_{s,l} - R_s\right) \tag{5}$$

where $\mathbf{x} = (x_s, \forall s \in \mathbf{S})$, $\mathbf{R} = (R_s, \forall s \in \mathbf{S})$, $\mathbf{X} = (X_{n,l}, \forall l \in \mathbf{L}, \forall n \in N(l))$; $N(l)$ is the set of input ports towards link l; $\boldsymbol{\lambda} = (\lambda_{n,l}, \forall l \in \mathbf{L})$ and $\boldsymbol{\mu} = (\mu_s, \forall s \in \mathbf{S})$ are *Lagrangian* multiplier vectors, which can also be interpreted as the "congestion prices" and "reliability prices", respectively. Thus, (5) can be reorganized as follows:

$$\begin{aligned} &L(\mathbf{x},\mathbf{R},\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu}) \\ &= \left(\sum_s U_s(x_s,R_s) - \sum_s \sum_{l\in L(s)} \lambda_{n,l}x_s - \sum_s \mu_s R_s\right) \\ &\quad + \left(\sum_l \sum_{n\in N(l)} \lambda_{n,l}X_{n,l} + \sum_l \sum_{n\in N(l)} \sum_{s\in S(n,l)} P_{s,l}\mu_s\right) \\ &\quad + \sum_s \mu_s \\ &= \sum_s \left\{U_s(x_s,R_s) - \mu_s R_s - \sum_{l\in L(s)} \lambda_{n,l}x_s\right\} \\ &\quad + \sum_l \sum_{n\in N(l)} \left\{\lambda_{n,l}X_{n,l} - \sum_{s\in S(n,l)} B_{n,l}\mu_s\right\} + \sum_s \mu_s. \end{aligned} \tag{6}$$

Therefore, the original optimization problem can be decomposed [17] as follows, where each user s controls his x_s and R_s, and each input port n towards link l controls $X_{n,l}$

$$\begin{aligned} Q(\boldsymbol{\lambda},\boldsymbol{\mu}) &= \max_{\substack{x_{\min}\le x_s \le x_{\max} \\ R_{\min}\le R_s \le 1 \\ X_{n,l}}} L(\mathbf{x},\mathbf{R},\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu}) \\ &= \max_{\substack{x_{\min}\le x_s \le x_{\max} \\ R_{\min}\le R_s \le 1}} \sum_s \left\{U_s(x_s,R_s) - \mu_s R_s - \sum_{l\in L(s)} \lambda_{n,l}x_s\right\} \\ &\quad + \max_{X_{n,l}} \sum_l \sum_{n\in N(l)} \left\{\lambda_{n,l}X_{n,l} - \sum_{s\in S(n,l)} B_{n,l}\mu_s\right\} \end{aligned} \tag{7}$$

$$\text{User } s: \max_{\substack{x_{\min}\le x_s\le x_{\max}\\ R_{\min}\le R_s\le 1}} U_s(x_s,R_s)-\mu_s R_s-\sum_{l\in L(s)}\lambda_{n(s,l),l}x_s \tag{8}$$

$$\text{Link } l: \max_{X_{n,l}} \sum_{n\in N(l)}\left\{\lambda_{n,l}X_{n,l}-\sum_{s\in S(n,l)}B_{n,l}\mu_s\right\}. \tag{9}$$

The *Lagrangian* multipliers are able to be obtained from following dual optimization problem

$$\min_{\substack{\lambda_{n,l}\ge 0\\ \mu_s\ge 0}} D(\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{10}$$

That problem can be solved through using the gradient projection method [22] as follows:

$$\begin{aligned}\lambda_{n,l}(t+1) &= \left[\lambda_{n,l}(t)-\Delta_\lambda\left(X_{n,l}-\sum_{s\in S(n,l)}x_s\right)\right]^+\\ \mu_s(t+1) &= \left[\mu_s(t)-\Delta_\mu\left(1-\sum_{l\in L(s)}B_{n,l}-R_s\right)\right]^+\end{aligned} \tag{11}$$

where Δ_λ and Δ_μ are step sizes; t is the time; $[z]^+ = \max\{z, 0\}$.

For each user s, its sending rate x_s can be determined by solving the problem in (8), shown as follows:

$$\begin{aligned}x_s &= \left[F^{-1}(0)\right]_{x_{\min}}^{x_{\max}}\\ \Rightarrow F(x_s) &= \frac{\partial\left(U_s(x_s,R_s)-\mu_s R_s-\sum_{l\in L(s)}\lambda_{n,l}x_s\right)}{\partial x_s}\\ &= \frac{\partial U_s(x_s,R_s)}{\partial x_s}-\sum_{l\in L(s)}\lambda_{n,l}=0\end{aligned} \tag{12}$$

where $[z]_a^b = \min\{\max\{z,a\},b\}$, and $F^{-1}(\cdot)$ is the inverse function of $F(\cdot)$.

Similarly, the reliability performance $R_{s,}$ requested by user s can be obtained as follows:

$$\begin{aligned}R_s &= \left[G^{-1}(0)\right]_{R_{\min}}^{1}\\ \Rightarrow G(R_s) &= \frac{\partial\left(U_s(x_s,R_s)-\mu_s R_s-\sum_{l\in L(s)}\lambda_{n,l}x_s\right)}{\partial R_s}\\ &= \frac{\partial U_s(x_s,R_s)}{\partial R_s}-\mu_s=0.\end{aligned} \tag{13}$$

As for the link side, it decides the optimal value of $X_{n,l}$ by solving (9). Directly solving multiple equations for $X_{n,l}$ is difficult, especially when the number of port is large. Thus, (9) can be solved by other ways like the gradient descent method.

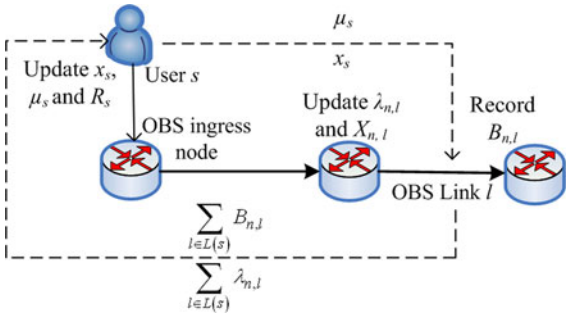

Fig. 3. The diagram of the proposed DPCC.

B. The Congestion Control Procedure

Through the theoretical analysis, we can see that DPCC is a distributed mechanism, in which those mentioned parameters interact with each other and will automatically converge to the optimal values to maximize the overall network utility after several iterations from any unorderly state. Its working principle is as shown in Fig. 3. All the ingress nodes update the $x_{s,}$, $\mu_{s,}$ and $R_{s,}$ belong to them in each iteration. Then, the latest updated x_s and μ_s can be loaded in the burst control packet (BCP) at the ingress node and sent out. All the links of the path from source to destination record the $\lambda_{n,l}$ and $B_{n,l}$ belong to each of them in real time. Then the latest produced $\lambda_{n,l}$ and $B_{n,l}$ are sent back to the corresponding ingress nodes before this iteration is over. The ingress nodes collect the $\sum_{l\in L(s)}\lambda_{n,l}$ and $\sum_{l\in L(s)}B_{n,l}$ to prepare for the updating of next iteration.

Because a higher/lower congestion price of $\sum_{l\in L(s)}\lambda_{n,l}$ corresponds to a lower/higher x_s, the user s determines its new sending rate x_s according to $\sum_{l\in L(s)}\lambda_{n,l}$. At the same time, the reliability price μ_s is updated based on the feedback $\sum_{l\in L(s)}B_{n,l}$ and former $R_{s,}$ according to (11). After that, the user knows what degree of a new reliability performance $R_{s,}$ he could expect based on the latest updated μ_s. For example, the user s could not expect a better end-to-end reliability $R_{s,}$ due to the higher reliability price $\mu_{s,}$. As for the link side, if the current x_s is over $X_{n,l}$ on link l, the link will increase the congestion price $\lambda_{n,l}$ to notify the rest of network that there is a congestion on that link. Link l also dynamically adjusts the $X_{n,l}$ according to current value of $\lambda_{n,l}$, and $\sum_{s\in S(n,l)}B_{n,l}\mu_s$ in order to alleviate or eliminate the congestion.

C. The Scalability of the DPCC

Due to the distributed nature, DPCC is scalable. It can be explained in detail from the two aspects as follows.

1) The information can be processed at each node in time as the network becomes bigger.

For each user s connected to the ingress node, it has three tasks to do as shown in Fig. 4(a).

Task1: update the data sending rate x_s according to (12).

Task2: update the reliability price μ_s according to (11).

Task3: decide what reliability performance R_s the user could expect according to (13).

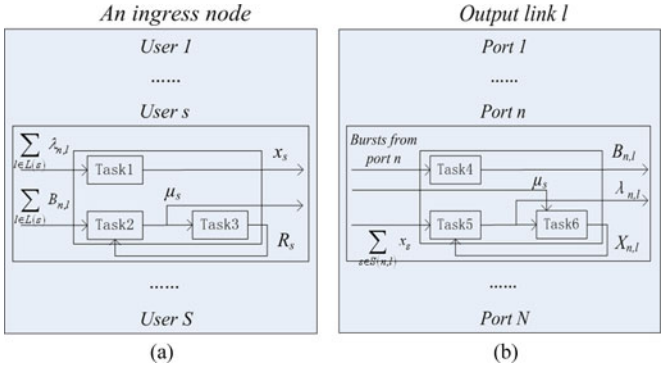

Fig. 4. The tasks at an ingress node (a); at an output link of a core node (b).

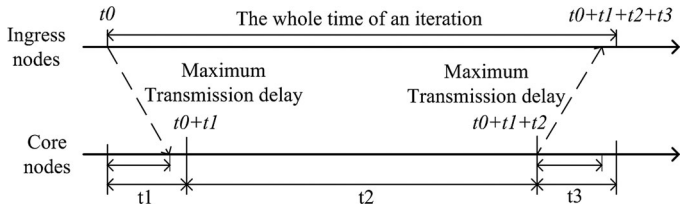

Fig. 5. The timing sequence of an iteration.

Task1 and Task2 can be done in parallel, then Task3 can be executed as soon as Task2 is done. In fact, for those users who have the same path, their x_s, μ_s and R_s are absolutely the same and it is enough to update any one of them. Hence, suppose there are *E* edge nodes in the network, an ingress node only needs to process *E-1* users' updating simultaneously at most which makes it quite feasible for the physical hardware like FPGA to perform such tasks.

As for each output link *l* of a core node, it also has three tasks to do as shown in the Fig. 4(b).

Task4: record the $B_{n,l}$.

Task5: update the congestion price $\lambda_{n,l}$ according to (11).

Task6: set the link parameters $x_{n,l}$ according to (9).

As with Task1 and Task2, Task4 and Task5 can be done in parallel, then Task6 can be executed at once after Task5. For the same output link *l*, the updating of $B_{n,l}$, $\lambda_{n,l}$ and $x_{n,l}$ of input port_n is independent of the updating of $B_{m,l}$, $\lambda_{m,l}$ and $X_{m,l}$ of input port_m, so they can be processed in parallel.

Not only total amount of information to be processed, but also the numbers of edge and core nodes that can process the information increase with the network size. Hence, as long as each node can process some of the data locally, the scalability of the DPCC in terms of the physical implementation is not a problem.

2) The parameters can converge within limited iterations as the network becomes bigger.

Since all the parameters interact with each other, when the network goes into the optimal state, they converge together. We take $\lambda_{n,l}$ as an example to evaluate how many iterations are needed for a network to converge. Generally speaking, the number of iterations needed to converge is decided by three metrics: the initial value, the steady value and the converging speed, as shown in

$$\text{iteration} = (\text{initial} - \text{steady})/\text{speed}. \tag{14}$$

The converging speed of $\lambda_{n,l}$ is $\Delta_\lambda \left(X_{n,l} - \sum_{s\in S(n,l)} x_s\right)$ according to (11). Since Δ_λ is 5×10^{-5} unit/Gb/s (please see [17] for further detail) and the ranges of $X_{n,l}$ and $\sum_{s\in S(n,l)} x_s$ are both from 1 to 20 Gb/s in the simulation, therefore $\Delta_\lambda \left(X_{n,l} - \sum_{s\in S(n,l)} x_s\right)$ varies between 10^{-3} and 0 (when the speed is 0, it means the network achieves optimal state at that time). In fact, its oscillation is becoming weaker and weaker exponentially since the difference between $X_{n,l}$ and $\sum_{s\in S(n,l)} x_s$ getting less and less as the iteration increases. Hence, according to the expression below, several values equals $p \times 10^{-3} \in (10^{-4}, 10^{-3})$, like 2×10^{-4} if a is 0.5 and the fastest speed at the beginning is 10^{-3}, may be reasonably taken as the average converging speed. If the difference between initial and steady values of $\lambda_{n,l}$ is not too great (0.1 unit more or less), the congestion price can converge in less than 1000 iterations. So can the reliability price. In addition, $x_{s,}$ and $R_{s,}$ which are controlled by those two prices, respectively, are also able to converge in a few hundreds of iterations

$$\left.\begin{array}{l} a < 1 \\ \sum a^n - \sum a^{n-1} < 0.001 \end{array}\right\} \Rightarrow n = \left\lceil \log_a^{0.001} \right\rceil$$
$$p = \frac{\sum a^n}{n} \approx \frac{1/1-a}{n} a < 1, n = \left\lceil \log_a^{0.001} \right\rceil. \tag{15}$$

In short, those corresponding parameters just need no more than a thousand iterations to converge if their initial values are set properly even when the network becomes bigger.

D. The Synchronization of the DPCC

As illustrated in Fig. 4, each ingress node iterates the algorithm based on the feedback information and then the core nodes update corresponding parameters according to the latest updated data produced at the ingress nodes. Since the transmission delays are different among nodes, it may make the convergence of the algorithm disordered if the DPCC iterates the algorithm at any ingress node as soon as the feedback information arrives. Hence, synchronization is needed to iterate the algorithm, which can be implemented as described below.

As shown in Fig. 5, suppose all the ingress nodes start to update their data at a moment *t*0 and send out the updated data as soon as possible. Since the transmission delays are various due to the different distances from ingress nodes to a certain core node (called *cn*), it is obvious that the updated data from ingress nodes cannot reach *cn* simultaneously. Thus, each core node prepares to get the new BLR of every input port at $t0 + t1$ where $t1$ should be longer than the transmission delay of the longest path in the network. In that case, we can assure all the flows that should pass over *cn* have arrived at *cn* before $t0 + t1$, which eliminates the influence of the transmission delay difference. Suppose the whole time used to get the BLR is $t2$, then all the core nodes begins to send their output data back to corresponding ingress nodes at $t0 + t1 + t2$. The time used for feedback transmission process is $t3$ (the same as *t1*), so all the ingress nodes are able to restart the next iteration at $t0 + t1 +$

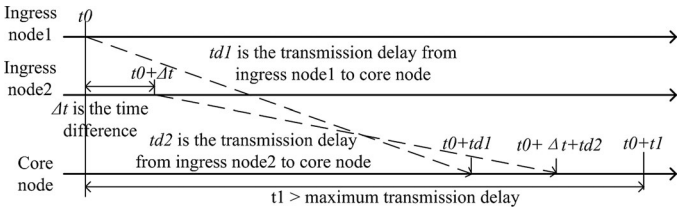

Fig. 6. The influence of the time difference in implementing parallel operation.

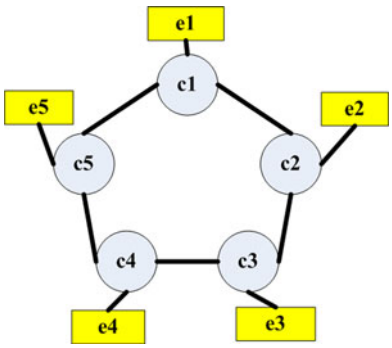

Fig. 7. The ring network.

$t2 + t3$. As a result, all the nodes iterate the algorithm per $t1 + t2 + t3$.

Let's take the NSFNet as an example here. The longest path is less than 5000 km when shortest path first (SPF) routing is adopted, which means the longest transmission delay is less than 17 ms. So it is sufficient to set $t1 = t3 = 20$ ms. $t2$ is used to record the BLR, which is at millisecond or second level. Besides, there inevitably exists time difference, also called synchronization accuracy problem, in implementing parallel operation among different nodes, i.e., all the ingress nodes do not begin to iterate algorithm exactly from $t0$ in reality, see Fig. 6. However, it is not a hard work for current technology to control the synchronization accuracy at microsecond level. In fact, the time difference Δt shown in Fig. 6 can be regarded as a virtual transmission delay difference whose value is only within microseconds. Thus, those little time differences can be neglected compared to $t1$ or $t2$.

IV. Simulation Results

We evaluate the performances of the proposed DPCC mechanism through two scenarios. The first is a five nodes ring network (see Fig. 7). It is a typical topology for a metropolitan optical network [23] and many complex networks can be considered as a combination of several simple topologies like this. The second is the NSFNet (see Fig. 9). The source-destination path is selected by the SPF routing strategy. Each link in those two networks has eight 10 Gb/s data channels and one 10 Gb/s control channel. We define each flow consists of 1000 users' burst data

$$U_s(x_s, R_s) = \alpha \frac{x_s^{1-\beta} - x_{\min}^{1-\beta}}{x_{\max}^{1-\beta} - x_{\min}^{1-\beta}} + (1-\alpha) \frac{R_s^{1-\beta} - R_{\min}^{1-\beta}}{R_{\max}^{1-\beta} - R_{\min}^{1-\beta}}. \tag{16}$$

The utility function for each user is a concave function of both data sending rate x_s and reliability performance $R_{s,}$, as shown in (16). Since the maximum value of $X_{n,l}$ should be 20 Mb/s and $\sum_{s \in S(n,l)} x_s \le X_{n,l}$, the maximum sending rate of single user cannot exceed 20 Mb/s. Thus, 20 Mbps is also the value of $x_{\max}$. $1 - \beta < 0$ and $0 < x_{\min} < x_{\max}$ are the two necessary points to make (16) a concave function, so we set $\beta = 1.1$ [24] and $x_{\min} = 1$ Mb/s in the simulation. The range of R_s is from 0.94 (when the end-to-end BLR is 6%) to 1. Parameter α can be tuned to give different weights to the sending rate and reliability performance ($0 \le \alpha \le 1$). For the sake of fairness, we set $\alpha = 0.5$ to make the sending rate and reliability have the equal weight in the simulation.

Four flows in the ring network are set in the simulation.

Flow1: e1 → c1 → c2 → c3 → e3.

Flow2: e2 → c2 → c3 → c4 → e4.

Flow3: e3 → c3 → c4 → c5 → e5.

Flow4: e4 → c4 → c5 → c1 → e1.

Flow2 has to contend with Flow1 and Flow3, and similarly Flow3 has to contend with Flow2 and Flow4, whereas Flow1 and Flow4 only need to contend with Flow2 and Flow3, respectively. Accordingly, Flow2 and Flow3 suffer from more congestion than Flow1 and Flow4, which affects the sending rate and reliability performances of the network.

The sending rate: since more congestion will lead to a higher congestion price, we can see that the congestion prices of Flow2 and Flow3 are higher than those of Flow1 and Flow4 when the network becomes stable, as shown in Fig. 8(a). The higher congestion price makes the users decrease their sending rates automatically, so the sending rates of Flow2 and Flow3 are lower than those of Flow1 and Flow4, as shown in Fig. 8(c).

The reliability: we have the following four observations from Fig. 8 (b), (d) and (e).

1) The end-to-end BLR of each flow increases with the rising of its sending rate, which shows there is a tradeoff between sending rate and the end-to-end BLR. 2) Although Flow1 and Flow4 have greater sending rates, their BLRs are lower than the BLRs of Flow2 and Flow3. This is because i) the streamline effect at a bufferless OBS node has a specialty like this: the greater the flow is, the less BLR it has [21]; ii) Flow2 and Flow3 experience more congestion than Flow1 and Flow2. 3) A higher BLR leads to a higher reliability price and a lower reliability. 4) R_{s} is the user's reliability anticipation obtained according to the reliability price μ_{s} in each updating, so it does not equal to the value of $1 - \sum_{l \in L(s)} B_{n,l}$ until the network achieves the optimal state, which means R_s is a fake reliability before its converging. R_{s} equals to $1 - \sum_{l \in L(s)} B_{n,l}$, the real reliability, when it is stable, making the reliability price $\mu_{s,}$ unchanged according to (11).

Fig. 8(f) shows the total user utility of the ring network. As can be seen, the proposed algorithm makes the total user utility converges and approaches to its maximum value after 600 iterations.

Although the ring network looks simple, given the fact that the DPCC and its scalability have been analyzed and illustrated independent of any given network or detailed form of utility

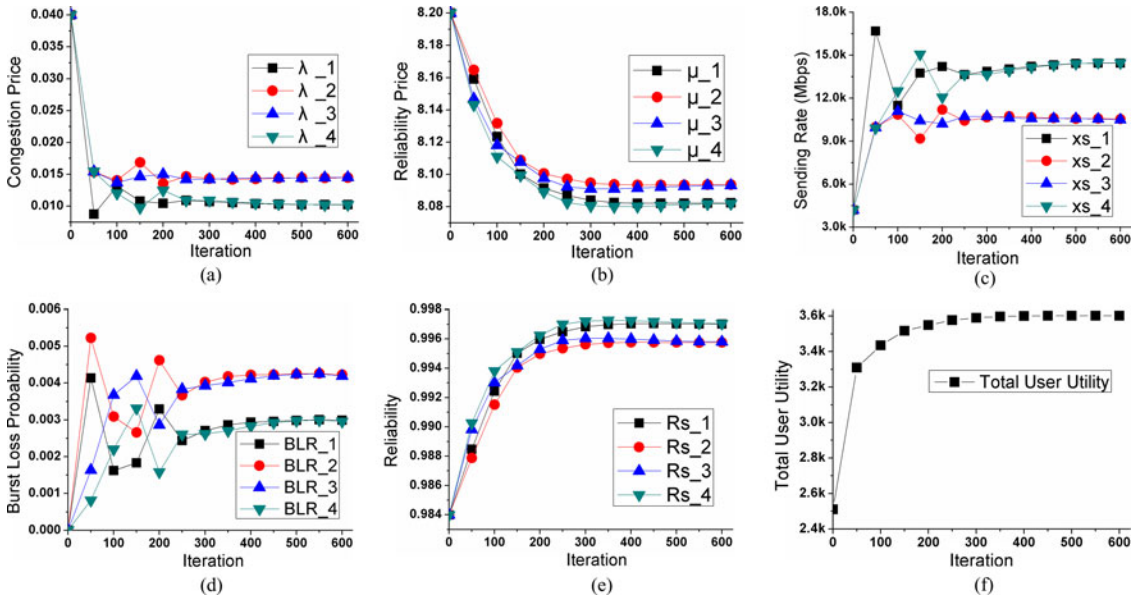

Fig. 8. The congestion price (a), the reliability price (b), the sending rate (c), the BLR (d), the reliability (e) and the total user utility (f) of the ring network.

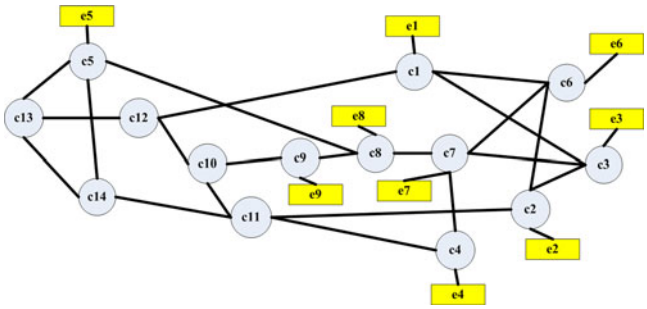

Fig. 9. The NSFNet.

function or parameters in last section, no matter what the network looks like (simple or complex) or various utility functions and parameters in it are adopted (similar or different), there is no effect on displaying the basic function of DPCC, which is deciding the sending rates of flows well for global optimization when the utility function, the parameter requirements and network topology are set previously.

To testify that point and make the DPCC more convincing, we further evaluate the performance of DPCC on a more complicated network, the NSFNet, with more traffic flows in contentions. Due to SPF routing, we find the worst contentions may occur at c7 node which has the highest nodal degree and typical representativeness since there will be at most four input links contend for an output link. Hence, we design following seven flows in order to: 1) let six flows from four input links contend for an output link at c7; 2) let multiple flows be streamlined more than once (e.g., Flow1–3 are streamlined at c6, Flow1–6 are streamlined at c7, Flow5–7 are streamlined at c8) and then observe the streamline effect under the control of DPCC.

Flow1: e1 → c1 → c6 → c7 → c8 → e8.
Flow2: e2 → c2 → c6 → c7 → c8 → e8.
Flow3: e6 → c6 → c7 → c8 → e8.
Flow4: e3 → c3 → c7 → c8 → e8.
Flow5: e4 → c4 → c7 → c8 → c9 → e9.
Flow6: e7 → c7 → c8 → c9 → e9.
Flow7: e5 → c5 → c8 → c9 → e9.

From the curves of Fig. 10, we can see although NSFNet is more complex, the same conclusions as the ring network can still be attained, which denotes the scalability of DPCC is unquestionable. 1) The flows undergo the same contentions have the same sending rate. For example, Flow1–3 are the same, Flow5–6 are the same. 2) Due to the streamline effect, the input port with smaller rate suffers more BLR at a single core node, which makes the BLR of Flow4 at c7 is nearly the same as the sum of BLRs of Flow1 at c6 and c7. 3) The flow experiences lighter (Flow7) and heavier (Flow5/6) collisions on its path will be allowed to have more and less sending rate, respectively. Both the sending rate x_s and reliability R_s of all flows converge to the optimal state automatically, and the utility achieves the maximum value well in the end.

The DPCC is used to achieve the optimal rate-reliability tradeoff of the whole network according to certain network optimization objects, which can be changed by adjusting α in the utility function. In the former simulation, we set $\alpha = 0.5$ to make the sending rate and reliability have the equal weight, which means the network pays the same attention to both sending rate and reliability. It is reasonable that the sending rates of some flows which experience more contentions are likely to be restricted to guarantee the overall network reliability. If the fairness of sending rate should be paid more attention in a network, one can increase the weight of the sending rate by increasing α. Fig. 11 shows the sending rate and the reliability of flows under different α. We can see the sending rate differences among different flows decreases at the cost of the decreasing the overall network reliability with the increase of α. One may note that reliability of Flow 7 almost does not change with α. This is because the variation of end-to-end BLRs of Flow7 is within 10^{-5} level

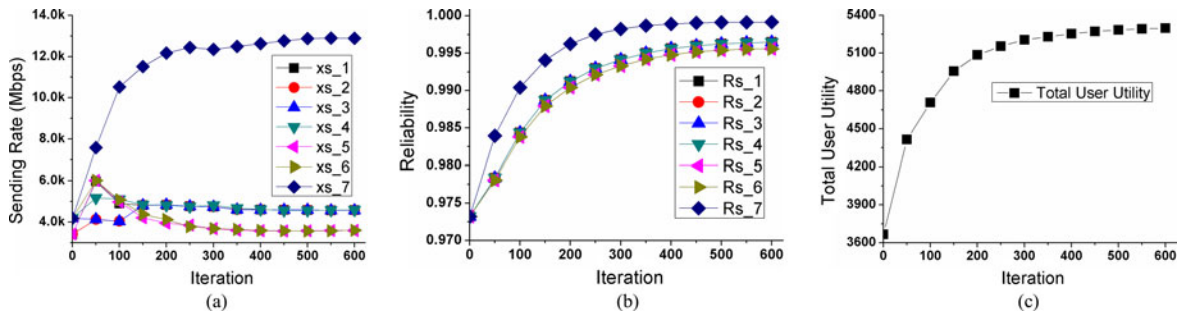

Fig. 10. The sending rate (a), the reliability (b) and the total user utility (c) of the NSFNet.

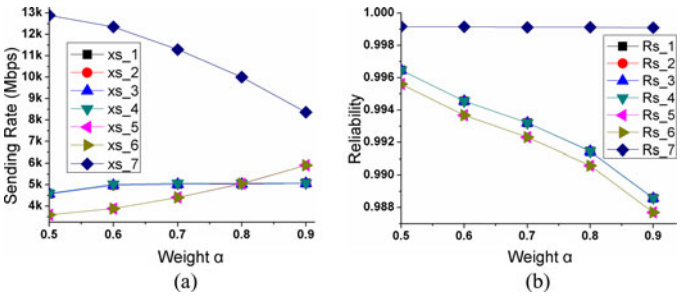

Fig. 11. The sending rate (a) and the reliability (b) when the weight α increases.

according to (3) when α increases from 0.5 to 0.9. Thus, the R_s_7 seems almost unchanged.

Resetting the parameter, like α, in the utility function will change the resulted values of flows, whereas it will not change the basic function of DPCC (i.e., to make the network obtain the optimal rate-reliability tradeoff). To further prove this feature, we carry out another simulation, in which the ($x_{\max}$, $x_{\min}$) of Flow2, Flow7 and other flows are assigned as (15, 1) Mb/s, (10, 1) Mb/s and (20, 1) Mb/s, respectively. We can see the whole network can still achieve convergence and optimal rate-reliability tradeoff under the control of DPCC from Fig. 12, although the performances of different flows changes correspondingly. Compared with Fig. 10 (a), when the $x_{\max}$ is 15 Mb/s, the sending rate of Flow2 is bigger than the case when $x_{\max}$ is 20 Mb/s, which complies with expressions (12) and (16) (i.e., the smaller the $x_{\max}$ is, the bigger the x_s is when other parameters are fixed). The sending rate of Flow7 is restricted at 10 Gb/s since x_s should not exceed its $x_{\max}$. Because the sending rates of flows vary not too much, their reliabilities look as if unchanged compared to Fig. 10(b) (e.g., according to (3), the variation of end-to-end BLR of Flow7 is within 10^{-4} level when its sending rate changes from 12.9 to 10.0 Gb/s).

In short, the DPCC is a mechanism to make the network achieve the optimal rate-reliability tradeoff for certain utility function and the parameter requirements, which is independent of any given form of the network topology, the utility function and the parameter requirements if their basic properties are satisfied according to the analysis in Section III. In other words, different object and requirements can be achieved under the control of DPCC by designating corresponding parameters such as α or ($x_{\max}$, $x_{\min}$) for different networks and application. In this paper, we just takes utility function (16) and those two networks as examples to prove the DPCC can arrange user rates rationally to globally optimize the network when the utility function, the parameter requirements and network topology are set previously. Though the utility function or parameters may not be most suitable for simulation networks, it is another topic beyond the scope of this paper.

Finally, as an optimization mechanism based on NUM, it is necessary to compare results from simulations with those of the primal problem since it is always not guaranteed that the solution of the dual problem by means of the *Lagrangian* relaxation is optimal for the primal one. It is hard to directly get the duality gap (the difference between the optimal values of primal problem and dual problem), but we can prove there is an upper bound of relative error (i.e., the ratio of duality gap to the optimal value of primal problem) and the bound is much small.

Suppose the optimal value of primal problem is $\sum_s U_s\left(x_s^{**}, R_s^{**}\right)$ and the optimal value of dual problem is

$$L\left(\mathbf{x}^*, \mathbf{R}^*, \mathbf{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) = \sum_s U_s\left(x_s^*, R_s^*\right) + \sum_l \sum_{n \in N(l)} \lambda_{n,l}^* \left(X_{n,l}^* - \sum_{s \in S(n,l)} {}^*_{x_s}\right) + \sum_s \mu_s^* \left(1 - \sum_{l \in L(s)} P_{s,l}^* - R_s^*\right).$$

Thus

$$\sum_s U_s\left(x_s^*, R_s^*\right) \le \sum_s U_s\left(x_s^{**}, R_s^{**}\right) \le L\left(\mathbf{x}^*, \mathbf{R}^*, \mathbf{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right)$$
$$\Rightarrow \text{Duality gap} = L\left(\mathbf{x}^*, \mathbf{R}^*, \mathbf{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) - \sum_s U_s\left(x_s^{**}, R_s^{**}\right)$$
$$\le L\left(\mathbf{x}^*, \mathbf{R}^*, \mathbf{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) - \sum_s U_s\left(x_s^*, R_s^*\right) \Rightarrow$$

$$\text{relative error} = \frac{\text{Duality gap}}{\sum_s U_s\left(x_s^{**}, R_s^{**}\right)}$$

$$\le \frac{L\left(\mathbf{x}^*, \mathbf{R}^*, \mathbf{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) - \sum_s U_s\left(x_s^*, R_s^*\right)}{\sum_s U_s\left(x_s^{**}, R_s^{**}\right)}$$

$$\le \frac{L\left(\mathbf{x}^*, \mathbf{R}^*, \mathbf{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) - \sum_s U_s\left(x_s^*, R_s^*\right)}{\sum_s U_s\left(x_s^*, R_s^*\right)}$$

$$= \frac{\left(\begin{array}{c} \sum_l \sum_{n \in N(l)} \lambda_{n,l}^* \left(X_{n,l}^* - \sum_{s \in S(n,l)} {}^*_{x_s}\right) \\ + \sum_s \mu_s^* \left(1 - \sum_{l \in L(s)} P_{s,l}^* - R_s^*\right) \end{array}\right)}{\sum_s U_s\left(x_s^*, R_s^*\right)}$$

$$= \text{ratio}.$$

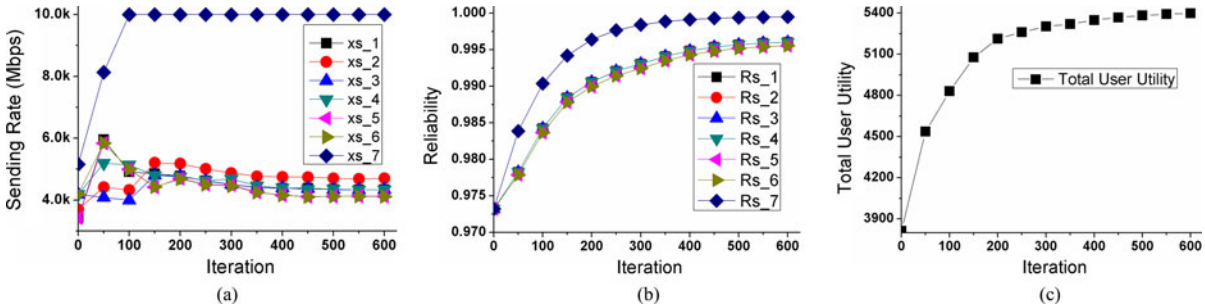

Fig. 12. The sending rate (a), the reliability (b) and the total user utility (c) of the NSFNet with different (x_{max}, x_{min}).

So if the "ratio" is much small, it means the duality gap is even smaller compared to the optimal value of primal problem. Through (11), we know $X^*_{n,l} - \sum_{s\in S(n,l)} x^*_s$ and $1 - \sum_{l\in L(s)} P^*_{s,l} - R^*_s$ are both approaching to zero when the congestion and reliability prices are nearly stable after iterations under the control of DPCC, which means the relative error is much small. In the NSFNet simulation containing seven flows, the upper bound of the relative error is 0.0283% after 600 iterations.

V. Further Discussion

Note that, DPCC mechanism adjusts the sending rate of users proactively according to the reliability performance (i.e., the loss ratio) of networks. So does TCP. However, the goal of the proposed DPCC mechanism for OBS networks is to achieve an optimal rate-reliability tradeoff, the TCP is to controls the rate based on the expression (17) below [25], [27]

$$x_s \approx \frac{1}{RTT_0 + 2T_b}\sqrt{\frac{N_T}{2bp_s}} = \frac{H}{\sqrt{p_s}} = \frac{H}{\sqrt{1-R_s}} \qquad (17)$$

where RTT_0 is the round trip time; T_b is the average assembly time; N_T is the average number of TCP packets in a burst; b is the number of packets that are acknowledged by a received ACK before increasing the sending window size [26] and it is typically 2; P_s is the end-to-end BLR for user s; and H is $\frac{1}{RTT_0+2T_b}\sqrt{\frac{N_T}{2b}}$. For example, if the average arrival rate of an OBS ingress node is about 20 Gb/s and the average length of a burst is 25 kB, then T_b is 10 us. If RTT_0 is 20 ms, T_b can be neglected in (17) and H is about 1 Mb/s. Considering $P_s = 1 - R_s$, as mentioned in Section II, there is a relationship between throughput X_s and burst reliability R_s according to (17).

Fig. 13 compares the different total user utilities of TCP, the DPCC with and without TCP in the ring network. In the DPCC with TCP, all parameters are updated according to DPCC except that the sending rate is updated according to (17). A larger H means a smaller network geography and/or longer bursts when $T_b << RTT_0$ according to (17). Fig. 13 demonstrates the following three features: 1) the proposed DPCC is able to achieve the optimal rate-reliability tradeoff through maximizing the total user utility compared with the other two mechanisms; 2) the utility of DPCC with TCP is greater than that of TCP

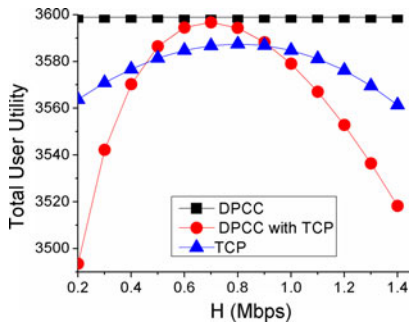

Fig. 13. The total user utility of TCP and the DPCC with and without TCP.

only when H falls into a certain optimal range since the sending rate of DPCC with TCP is closer to the expected optimal rate obtained in DPCC in that range. In our simulation, the optimal range for H is from 0.5 to 0.9 Mb/s; 3) DPCC with TCP can reach a larger utility sooner than TCP as H increases from a smaller value.

VI. Conclusion

We have proposed a distributed dual price-based congestion control (DPCC) mechanism for OBS networks based on the idea of NUM. It achieves an optimal rate-reliability tradeoff by controlling the users' data sending rate and the end-to-end reliability performance of burst transmission by setting and adjusting both the congestion and reliability prices according to the congestion state of the OBS network. The performance of the proposed DPCC mechanism has been observed and analyzed through simulations. The results show that 1) the total user utility, also the total users' satisfaction degree of the network, can be maximized after a few iterations and will remain to be in the maximized state; 2) traffic with a larger/smaller end-to-end BLR will end up with a lower/higher sending rate due to the effects of the congestion and reliability prices, which allows the network to achieve the optimal rate-reliability tradeoff; 3) compared with TCP, the proposed DPCC mechanism can always achieve a larger utility and an optimal rate-reliability tradeoff.

Due to its capability to arrange proper network traffic and resources, it is expected to play important role as a new routing scheme for reducing overall contention in a future optical network.

REFERENCES

[1] C. Qiao and M. Yoo, "Optical burst switching (OBS)—A new paradigm for an optical Internet," *J. High Speed Netw.*, vol. 8, no. 1, pp. 69–84, Jan. 1999.

[2] G. Wu, T. Zhang, J. Chen, X. Li, and C. Qiao, "An index-based parallel scheduler for optical burst switching networks," *J. Lightw. Technol.*, vol. 29, no. 17, pp. 2766–2773, Sep. 2011.

[3] G. B. Figueiredo and N. L. S. Da Fonseca, "Channel reusability for burst scheduling in OBS networks," *J. Photon. Netw. Commun.*, vol. 26, no. 2–3, pp. 84–94, 2013.

[4] T. Zhang, G. Wu, X. Li, and J. Chen, "A High speed scheduler with a novel scheduling algorithm for optical burst switching networks," *J. Lightw. Technol.*, vol. 31, no. 18, pp. 2844–2850, Sep. 2013.

[5] V. Kavitha and V. Palanisamy, "New burst assembly and scheduling technique for optical burst switching networks," *J. Comput. Sci.*, vol. 9, no. 8, pp. 1030–1040, 2013.

[6] L. Wang, Y. Chen, and M. Thaker, "Virtual burst assembly—A solution to out-of-sequence delivery in optical burst switching networks," in *Proc. IEEE Global Telecommun. Conf.*, 4698277, 2008, pp. 2617–2622.

[7] K. Seklou, A. Sideri, P. Kokkinos, and E. Varvarigos, "New assembly techniques and fast reservation protocols for optical burst switched networks based on traffic prediction," *Opt. Switch. Netw.*, vol. 10, no. 2, pp. 132–148, 2013.

[8] A. Guan, B. Wang, and T. Wang, "Contention resolution and burst assembly scheme based on burst segmentation in optical burst switching networks," *Optik*, vol. 124, no. 14, pp. 1749–1754, 2013.

[9] F. Farahmand and J. Jue, "A feedback-based contention avoidance mechanism for optical burst switching networks," in *Proc. 3rd WOBS*, vol. 1, Oct. 2004, pp. 15–20.

[10] S. Kim, B. Mukherjee, and M. Kang, "Integrated congestion-control mechanism in optical burst switching networks," in *Proc. IEEE Global Telecomm, Conf.*, 1578011, 2005, pp. 1973–1977.

[11] S. Kim, Young-Chou, B.-Y. Yoon, and M. Kang, "An integrated congestion control mechanism for optimized performance using two-step rate controller in optical burst switching networks," *Comput. Netw.*, vol. 51, no. 3, pp. 606–620, 2007.

[12] S. Low, "A duality model of TCP and queue management algorithms," *Trans. Network.*, vol. 11, no. 4, pp. 525–536, Jan. 2003.

[13] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[14] F. P. Kelly, A. Maulloo, and D. Tan "Rate control for communication networks: Shadow prices, proportional prices and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[15] W.-S. Park, M. Shin, H.-W. Lee, and S. Chong, "Joint congestion control and burst contention resolution in optical burst switching networks," in Proc. IEEE Global Telecommun. Conf., 2007, pp. 2209–2214.

[16] W.-S. Park, M. Shin, H.-W. Lee, and S. Chong, "A joint design of congestion control and burst contention resolution for optical burst switching networks," *J. Lightw. Technol.*, vol. 27, no. 17, pp. 3820–3830, Sep. 2009.

[17] S. H. Low and D. E. Lapsley, "Optimization flow control—Part I: Basic algorithm and convergence," *Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[18] K. Ronasi, A. H. Mohsenian-Rad, V. W. S. Wong, S. Gopalakrishnan, and R. Schober, "Reliability-based rate allocation in wireless inter-session network coding systems," in *Proc. IEEE Global Telecommun. Conf.*, 5425445, 2009, pp. 1–6.

[19] J. W. Lee, M. Chiang, and A. R. Calderbank, "Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization," *J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 962–976, May 2006.

[20] J. W. Lee, M. Chiang, and A. R. Calderbank, "Network utility maximization and price-based distributed algorithms for rate-reliability tradeoff," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–13.

[21] M. H. Phung, D. Shan, K. C. Chua, and G. Mohan, "Performance analysis of a bufferless OBS node considering the streamline effect," *Commun. Lett.*, vol. 10, no. 4, pp. 293–295, Apr. 2006.

[22] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1995.

[23] J. M. Finochietto, J. Aracil, and Á. Ferreiro, J. P. Fernández-Palacios Giménez, and Ó. G. de Dios, "Migration strategies toward all optical metropolitan access rings," *J. Lightw. Technol.*, vol. 25, no. 8, pp. 1918–1930, Aug. 2007.

[24] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *Trans. Network.*, vol. 8, no. 5, pp. 556–567, Oct. 2000.

[25] X. Yu, C. Qiao, Y. Liu, and D. Towsley, "Performance evaluation of TCP implementations in OBS networks," CSE Department, SUNY at Buffalo, Buffalo, NY, USA, Tech. Rep., 2003-13, 2003.

[26] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM Conf. Appl. Technol. Archit. Protocol Comput. Commun.*, 1998, pp. 303–314.

[27] X. Yu, C. Qiao, and Y. Liu, "TCP implementations and false time out detection in OBS networks," in *Proc. IEEE Annu. Joint Conf. Comput. Commun. Soc.,* 2004, pp. 774–784, vol. 2.

Tairan Zhang received the B.S. degree from China University of Mining and Technology, Xuzhou, China, in 2005 and M.S. degree from East China Normal University, Shanghai, China, in 2008, respectively. He is currently working toward the Ph.D. degree in optical communication at the Shanghai Jiao Tong University, Shanghai, China. His research focuses on the modeling and experimental study of optical burst switching networks.

Wei Dai received the B.S. and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2009, respectively. He is currently working toward the Ph.D. degree with the Networked Systems, University of California, IR, CA, USA.

Guiling Wu received the B.S. degree from Haer Bing Institute of Technology, Heilongjiang, China, in 1995, and the M.S. and Ph.D. degrees in optical electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively. He is currently an Associate Professor with the State Key Laboratory of Advanced Optical Communication Systems and Networks, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include optical networking and high-speed optical signal processing.

Xinwan Li received the M.S. degree from Shanghai University, Shanghai, China, in 1993, and Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2005. Since 1993, he has been with Shanghai Jiao Tong University, Shanghai, China, where he is currently a Professor. From 1997 to 1998, he was with Essex University, U.K., as a Research Assistant. In 2001, he joined OPCOM, Inc., as an Engineer and as a Visiting Professor of Chonbuk National University in 2007. His main research interest includes optical switching technologies and advanced optical fiber components. He is a Senior Member of the IEEE Photonics Society and the Chair of IEEE Communications Society shanghai chapter.

Jianping Chen received the B.S. degree from Zhejiang University, Hangzhou, China, in 1983, and the M.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1986 and 1993, respectively. He is currently a Professor with the State Key Laboratory of Advanced Optical Communication Systems and Networks, Department of Electronic Engineering, Shanghai Jiao Tong University. His main research interests include photonic devices and signal processing, optical networking, and sensing optics. He is also a Principal Scientist of 973 Project in China.

Chunming Qiao (F'10) is currently a Professor with the State University of New York at Buffalo, NY, USA, where he directs the Laboratory for Advanced Network Design, Evaluation and Research. He pioneered the research on optical burst switching and integrating cellular and ad hoc relaying technologies. His research has been supported by a number of National Science Foundation grants, including two Information Technology Research awards, and by a number of major networking research and development organizations. He has authored or coauthored more than 250 papers in leading technical journals and conference proceedings, authored several book chapters, and given dozens of keynote speeches, tutorials, and invited talks.