This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Integrated Congestion-Control Mechanism in Optical Burst Switching Networks

Sungchang Kim*, Biswanath Mukherjee**, and Minho Kang*
* Optical Internet Research Center, Information and Communications University, Daejeon, Korea
** Department of Computer Science, University of California, Davis, CA 95616, USA
E-mail : pluto@icu.ac.kr, mukherje@cs.ucdavis.edu, mhkang@icu.ac.kr

Abstract -- **Optical burst switching (OBS) is a promising solution to implement the optical internet backbone. However, the lack of adequate congestion-control mechanisms may result in high burst loss. Schemes such as fiber delay line (FDL), wavelength conversion, and deflection routing to reduce burst collision are unable to prevent the network congestion effectively. To address this problem, we propose and investigate a global solution, called *integrated congestion-control mechanism* (ICCM), for OBS networks. ICCM, which combines congestion avoidance with recovery mechanism, restricts the amount of burst flows entering the network according to the feedback information from core routers to edge routers to prevent network congestion. Also, a flow-policing scheme is proposed to intentionally drop the overloaded traffic with a certain probability at a core router to support fairness among flows. Moreover, the transmission rate of each flow is controlled to achieve optimized performance such as maximizing throughput or minimizing loss probability using a two-step rate controller at the edge router. Simulation results show that ICCM effectively eliminates congestion within the network and that, when combined with a flow-policing mechanism, the fairness for competing flows can be supported while maintaining effective network performance.**

I. INTRODUCTION

Until now, several solutions have been proposed to resolve the contention problem in OBS networks [1][2]. However, in all proposed solutions, these mechanisms offer only local treatment rather than a network-wide global solution which controls the network traffic volume properly.

Congestion in an OBS network is a state in which performance degrades due to the saturation of network resources such as WDM links, wavelength converters, and FDLs. Adverse effects resulting from such congestion include the high loss probability of data bursts, waste of wavelength resources, and possible network collapse. Network congestion is becoming a real threat to the growth of communication applications and their QoS requirements.

Current network-control mechanisms need an end-to-end overview of various traffic flows rather than the information that is purely local to the individual nodes. Along this line, end-to-end congestion control in OBS requires some form of feedback information from the congested core nodes to the ingress source nodes of the data bursts, so that they can adjust their rates of injecting data into the network according to the available bandwidth in the network. Additionally, in an OBS network, the edge routers have all of the intelligence while the core nodes have simple cut-through switching function, so that the feedback information of network state from core to edge is essential with respect to congestion control.

So far, there has been little consideration about feedback-based congestion-control mechanism in OBS networks. In [3], a feedback-based contention-avoidance mechanism was proposed in OBS networks. The core routers send explicit feedback messages to edge routers requesting them to reduce the transmission rate on a congested link. The performance results show that the explicit feedback mechanism can reduce the loss probability, and increase network utilization. However, due to the bursty and unpredictable nature of traffic, this preventive approach is not sufficient to control the congestion, and additional reactive controls may be necessary in the network. It is highly probable that, if the sources sharing the bandwidths of the network link have high QoS requirements, some peak-rate allocation may be the only solution to an approach entirely based on preventive control.

In this paper, we propose an *integrated congestion-control mechanism* (ICCM) which takes preventive and reactive controls into consideration in an integrated manner in OBS networks. In this mechanism, we combine dynamic access control which prevents overload of the network using leaky-bucket shapers in ingress routers based on explicit feedback mechanism, and a flow-policing scheme which recovers from congestion when the load exceeds some predefined thresholds using intentional burst dropping in core routers.

The main objective of this paper is to develop an efficient congestion-control mechanism in which the network overloads are minimized or eliminated to improve network utilization. Also, we consider the fairness performance between competing flows. In order to provide different fairness property, we consider a weighted max-min fair drop scheme that can run at the core routers to meet the fairness requirements. Furthermore, we consider performance optimization which is related to the rate-decision problem.

II. INTEGRATED CONGESTION-CONTROL MECHANISM

A. Function of Core Router

Core routers in an OBS network are responsible for four main jobs: transparent cut-through switching of data bursts, analyzing (monitoring) incoming bursts, enforcing the intentional burst drop of sustained overloads to keep the traffic volume within a controllable level, and providing feedback information to the edge nodes.

Burst control packets (BCP) sent by ingress routers arrive at the control input port of a core router and are first classified by flows. Flow-classification policy is to examine the BCP's ingress and egress addresses. After classifying bursts into flows, each flow's incoming rate is monitored using a rate-estimation algorithm such as an exponential moving average rate from the total bursts received in a certain interval of time. These monitored input rates are collected at the feedback controller, which generates *feedback control packets* (FCP) and distributes to all edge routers whenever the predefined timer expires.

The flow-policing mechanism intentionally drops some of the BCPs to control the congestion whenever the total arrival rate exceeds a certain threshold. After the flow-policing mechanism, the BCPs try to reserve a proper wavelength for

the corresponding data burst using various scheduling algorithms such as Horizon and LAUC-VF (Latest Available Unused Channel - Void Filling) [4]. If the BCPs can not find any available wavelength, they can be dropped.

B. Function of Edge Router

The edge routers are responsible for several jobs: traffic aggregation, generation of BCPs and data bursts, offset-time[1] calculation, and deciding on the optimal token rate of leaky bucket. Each ingress router contains flow classifier, burst-generation queue, leaky-bucket traffic shaper, feedback controller, and rate controller. First, arriving BCPs are classified by their destination egress node, and then they move into the corresponding burst-generation queue. In order to transmit a data burst, which has finished its assembly process and which is at the head of the queue to be transmitted, the burst should get a token from the token pool.

Token rates are decided at the rate controller based on the FCPs at every core routers. For a given path, an optimal token rate is determined by the most-congested node on the path. We present a rate-decision algorithm and its performance-optimization issues in Section IV.

C. Feedback Control Packet

The FCPs may be broadcast to all ingress nodes periodically or when the offered loads change significantly with time. They are also transmitted with higher priority than BCPs to ensure reliable transmission without delay and loss.

Contained within the FCP generated at a core router are a header, and a list of observed arrival rates for each flow. The list of flow specification indicates to an edge router the identities of active flows originating at each ingress router. A flow specification is a value uniquely identifying a flow which is assigned using ingress and egress addresses. The core router adds a flow to its list of active flows whenever a burst from a new flow arrives; it removes a flow when the flow becomes inactive.

III. FLOW-POLICING SCHEME

This section presents a detailed formulation of the flow-policing mechanism which effectively prevents overload situations when the dynamic access control does not properly handle congestion due to delayed response of feedback mechanism and fluctuations of input traffic. To do this, we consider an intentional dropping scheme as a flow-policing mechanism, which should guarantee that all flows are treated fairly when they compete for a common bottleneck link.

To address this concern, we define two different fairness properties: rate fairness and distance fairness. Rate fairness indicates that the bandwidth of each flow which competes for the same output link is allocated fairly according to its offered rate. The distance fairness represents that the burst should be treated fairly with respect to hop count of each burst, since flows which take more hops may get lower throughput due to the burst loss at intermediate routers.

To describe the proposed flow-policing mechanism, the following assumptions and variables are defined. We consider that M edge routers transmit and receive data bursts through N core routers. Let $r_i^{sum} = \sum_{j=1}^{P} r_i^j$ represent the total offered load of core router i, where r_i^j is the offered load of flow j and P is is the total number of flows that traverse through core router i. If the total offered load is less than the output link capacity of core router i, C_i, then no bursts are intentionally dropped. Otherwise, an overloaded core router i should intentionally drop of at least $r_i^{sum} - C_i$ of the bursts to maintain utilization equal to or below the output link capacity. Each data burst in flow j is intentionally dropped at core router i, with probability D_i^j, to statistically enforce the desired offered load. In our proposed formulation, the intentional-drop probability D_i^j that an overloaded core node assigns to each flow is based on the fairness criteria, such as rate and distance fairness.

To describe the intentional-drop policies formally, we define several parameters as follows.

r_i^j : Arrival rate of flow j at core router i, normalized to the network's fastest link, to give a dimensionless utilization between 0 and 1 where $1 \le j \le P$.

r_i^1 : Smallest arrival rate at core router i among active flows.

h_i^j : Number of hops experienced by flow j when it arrives at core router i, normalized to the maximum end-to-end hop count, such that this value is between 0 and 1.

h_i^1 : Number of hops of r_i^1.

β : Distance factor which controls the effect of hop count for guaranteeing the distance fairness.

In our model, to inform the h_i^j at a the core router, we assume that the BCP has an optional field which includes the number of hops already passed. We also assume that each ingress node can perform explicit routing based on feedback information, so that the ingress node knows the hop count to any core router.

A. Weighted Max-Min Fair Drop (WMFD) Scheme

Max-min fair drop gives the most-poorly-treated flow (i.e., the flow which transmits at the lowest rate) the largest possible share of bandwidth, while not wasting any network resources. To support distance fairness, the weighted max-min fair drop (WMFD) scheme adopts a new weight which is $w_i^j = (h_i^j)^\beta / \sum_{k=1}^{P} (h_i^k)^\beta$. To allocate D_i^j, the following algorithm is used in each core router.

Weighted Max-min Fair Drop Algorithm

1) Calculate $r_i^{sum} = \sum_{j=1}^{P} r_i^j$.

- If $r_i^{sum} \le C_i$, then $D_i^j = 0$ for all j, $1 \le j \le P$, exit.

- If $r_i^{sum} > C_i$, then $r_i^{sum} - C_i$ should be intentionally dropped.

2) Pick the smallest rate r_i^1 among P and calculate w_i^1

- If $r_i^1 \le w_i^1 C_i$, then $D_i^1 = 0$.

- If $r_i^1 > w_i^1 C_i$, then $D_i^1 = \{r_i^1 - (w_i^1 C_i)\} / r_i^1$.

3) Set $P \leftarrow P - 1$, $C_i = C_i - r_i^1 (1 - D_i^1)$.

4) If $P > 0$, then Go Step 2).

We can observe that, if $\beta = 0$, WMFD is equal to max-min fair drop. But, if $\beta > 0$, the larger the β value is, the more penalty is given to the bursts which are transmitted with a small number of hops.

[1] Offset time is a time gap between BCP and corresponding data burst which compensates for BCP processing time in OBS core routers.

B. Analysis of Loss Probability

In this subsection, we present an analysis of the loss probability for each flow at a core router. The core router which adopts a flow-policing scheme can be developed as a two-stage loss model. At the first stage, bursts are intentionally dropped with probability D_i^j at the flow-policing stage. At the second stage, bursts which survived the first stage are scheduled at the proper wavelength based on various scheduling algorithms such as First Fit, Random, Horizon, and LAUC-VF. Under the assumption that each flow arrives in a Poisson stream, services with exponentially-distributed service time, and contains k wavelengths in a fiber, the loss at the scheduling stage, B_i^j, can be evaluated through Erlang's loss formula for the loss probability of an *M/M/k/k* system [5]:

$$B_i^j = B(k, \rho_i) = (\rho^k / k!) / \sum_{i=0}^{k} (\rho^i / i!) \quad (1)$$

where ρ_i is the total offered load at core router i. Let L_i^j be the loss probability of flow j at core router i; then, we should verify two different conditions:

Case 1) Normal condition (Congestion free)

$D_i^j = 0$, $B_i^j = B(k, r_i^{sum})$; therefore, $L_i^j = B_i^j$.

Case2) Congested condition

D_i^j = Followed by WMFD, $B_i^j = B(k, C_i)$.

Therefore, $L_i^j = 1 - (1 - D_i^j)(1 - B_i^j)$.

If we assume that flow j has traveled through N core routers, then the end-to-end loss probability and throughput are given by:

$$L_j = 1 - \prod_{k=1}^{N} (1 - L_k^j) \quad (2)$$

$$Th_j = r_{source}^j (1 - L_j) \quad (3)$$

IV. RATE-DECISION MECHANISM

The rate-decision mechanism regulates the rate at which each flow is allowed to enter the network. Its primary goal is to converge on a set of per-flow transmission rates that prevents congestion due to overloads. It also attempts to lead the flows to a state of optimized performance.

On the ingress-router side, upon receipt of the FCP, the rate controller determines the transmission rate of a burst by changing its leaky-bucket parameters. This can be done by either changing the token rate in the bucket, or by changing the bucket size. In our model, we use the former since the latter has a secondary effect in throttling the traffic.

The design process of the rate controller can be divided into two main parts: online part and offline part. The online part is responsible for the real-time control of the network, and it prevents the network from congestion immediately. The offline part further tunes the token rate which is derived from the online part with the optimization criteria. We adopt a simulated annealing (SA)-based algorithm as the optimization tool for the offline algorithm.

In the online part, a token rate for each flow is chosen for the most-congested link (the *bottleneck link*) on the path. If we assume that flow j traverses N intermediate core routers, the online algorithm first calculates the optimum transmission rate $r_{opt\_i}^j (1 \le i \le N)$ of each intermediate core router along the path. We can define the optimum transmission rate $r_{opt\_i}^j$ to be the weighted max-min fair share rate of flow j in core router i. Once we obtain all $r_{opt\_i}^j$ along the path, the token rate of flow j, R_j, can be determined as follows:

$$R_j = r_{opt}^j = \min\{r_{opt\_i}^j \mid 1 \le i \le N\}$$

A. SA-based Optimum Rate-Decision Algorithm

Each ingress router chooses token rates using the online algorithm for instantaneous response. In the offline part, many optimization criteria are possible, including maximizing throughput, and minimizing loss probability. In this subsection, we only deal with maximizing throughput of each flow. In fact, the end-to-end throughput is determined by the bottleneck link along the path. However, the flows can experience burst loss at each hop due to intentional dropping or contention during scheduling, resulting in reduced throughput. For example, even though the online algorithm calculates the token rate based on the bottleneck link, the arriving bursts at the bottleneck link are much smaller than the optimum rate due to burst loss at intermediate routers. In order to achieve optimized performance, we propose the SA-based optimum rate-decision algorithm for the offline part.

A numerical implementation of SA consists of a data structure for the state or solution space of the problem, a probability distribution on the transition between states, a temperature variable[2], and an optimization function defined on states. We now detail our assignments of these ingredients to the problem of maximizing the throughput of each flow.

1) *Sate Space:* We assume that there are M edge routers in the network, and each ingress router has *M-1* leaky buckets, one for every other ingress/egress routers. Our solution space will be the set of all *(M-1)*-tuples of token rates:

$$\overline{R} = (R_1, R_2, ..., R_{M-1})$$

with the constraint $r_{opt}^j \le R_j \le r_{opt\_first}^j$ for ($1 \le j \le M-1$), where r_{opt}^j is the token rate which is determined by the online algorithm, and $r_{opt\_first}^j$ is the optimum transmission rate of the first core router along the path.

2) *Objective Function*: Associated with every token rate vector $\overline{R}$ in the state space, we have $Th_{agg} = z(\overline{R}) = \sum_{j=1}^{M-1} Th_j$, where Th_{agg} is the aggregated network throughput for an ingress router and throughput Th_j as determined from Eqn. (2). Therefore, our objective function can be represented by:

$$Th_{agg}^{best} = Max\{z(\overline{R})\}$$

3) *Transition Probability Distribution*: The transition from state to state is affected by means of two processes. The first, the generation process, generates new trial states as a function of the known present state, as follows:

Generate ($\overline{R}_{best}$);

$k \leftarrow 1 + (\text{int})[(M-1) \times rand()]$;

[2] Temperature variable is an important parameter which directly impacts the efficiency of the algorithm. If the temperature is decreased too fast (rapid cooling process), the solution could be stuck at a local optima, otherwise (for a slow cooling process), the running time of the algorithm could get significantly increased.

$R_k \leftarrow r_{opt}^k + [(r_{opt\_first}^k - r_{opt}^k) \times rand()];$

$return\ \overline{R}_{trial} = \{R_1, R_2, ..., R_{M-1}\}$

where $\overline{R}_{best}$ is the best solution for the token vector up to now, and $\overline{R}_{trial}$ is the next trial token vector. Having generated a trial token vector $\overline{R}_{trial}$ as a perturbation of the present token vector $\overline{R}_{best}$, it is accepted or rejected as the next state of the process probabilistically in accordance with the Boltzmann distribution law. The formal description of replacement process is as follows:

$Replace\ (\overline{R}_{best}, \overline{R}_{trial})$

$Th_{agg}^{trial} \leftarrow z(\overline{R}_{trial})$;

$\Delta Th_{agg} \leftarrow Th_{agg}^{trial} - Th_{agg}^{best}$;

$if\ \Delta Th_{agg} \geq 0\ return\ \overline{R}_{trial}$;

$if\ (rand() < \exp((\Delta Th_{agg})/T))\ return\ \overline{R}_{trial}$;

$return\ \overline{R}_{best}$.

4) Temperature Management: In our work, we apply the hyperbolic cooling process ($T = d/(d+k)$) which performs very rapidly cooling. The parameter d should be at least as large as the height of all non-global maxima and k is an iteration count.

The complete SA-based optimum rate-decision algorithm for offline part is given as follows:

SA-based Optimum Rate-Decision Algorithm

$T \leftarrow 1$;

$\overline{R}_{best} \leftarrow Random\_initialization()$;

$Th_{agg}^{best} \leftarrow z(\overline{R}_{best})$;

$k \leftarrow 1$;

do

$\overline{R}_{trial} \leftarrow Generate(\overline{R}_{best})$;

$\overline{R}_{best} \leftarrow Replace(\overline{R}_{best}, \overline{R}_{trial})$;

$Th_{agg}^{best} \leftarrow z(\overline{R}_{best})$;

$T \leftarrow d/(d+k)$;

$k \leftarrow k+1$;

while $(T > T_{stop})$.

V. Illustrative Numerical Examples

We now present results from simulation experiments, each of which is designed to verify a different aspect of ICCM performance. The simulation is event-driven, and is simulated at the burst level. We assume that FCPs are never lost in the network. The network configurations used for the simulations are shown in Figs. 1, and 5. Default simulation parameters are listed in Table I. Without loss of generality, we assume that the normalized capacity of each output link is 1. In Fig. 1, Flows 1 and 2 generate their data bursts with offered load 0.3 at time=0 second. Flows 3, 4, and 5 start their transmission at 5, 10, and 15 seconds with offered loads 0.6, 0.5, and 0.4, respectively.

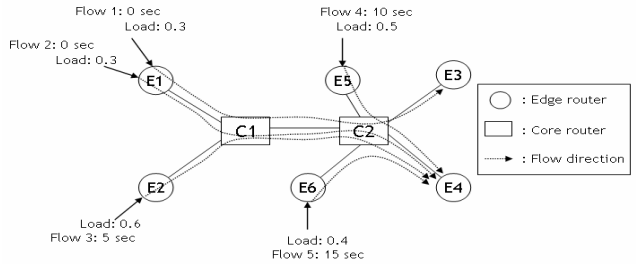

Fig. 1. Network scenario for congestion control.

TABLE I
Default Simulation Parameters

Simulation Parameters	Value
Data burst size	200 μs
Offset time	20 μs
Number of wavelengths in a fiber	8
Bandwidth of each wavelength	10 Gbps
Feedback interval	5 sec
Offline algorithm calculation time	1 sec
Rate-estimation interval	1 sec
Cooling parameter d	10

A. Preventing Congestion in the Network

Our first result shows ICCM's ability to prevent congestion in the network. Figure 2 shows total offered load at the core routers versus simulation time. Without any congestion control, core router C1 experiences severe overload after Flow 3 starts its transmission. On the other hand, ICCM effectively reduces the total offered load which is below or equal to the capacity of the output link, because each flow properly limits its transmission rate to prevent congestion whenever the feedback information is updated. The congestion of core router C2 can be observed after Flow 4 starts its transmission as shown in Fig. 2. Starting up of new flows creates small peaks and deviations from capacity during a transient phase, but following a short period of reaction and rate adjustment, the total offered load stabilizes to the capacity which the core router can handle.

B. Effectiveness of SA-based Rate-Decision Algorithm

If we consider the performance difference between the online and the offline algorithms, the online algorithm reduces the total offered load of C1 below the capacity after 20 sec as shown in Fig. 2, since the bottleneck link for Flows 2 and 3 has been moved from link (C1-C2) to link (C2-E4). Therefore, Flows 2 and 3 adjust their token rates based on information on the most-congested link (C2-E4) rather than link (C1-C2), resulting in low link utilization of link (C1-C2). When only the online algorithm is used, offered loads of Flows 2 and 3 into C2 are smaller than the optimal values, due to their burst loss in C1, so that Flows 2 and 3 can not achieve maximum throughput (fair share bandwidth). As a result, the throughputs of Flows 2 and 3 are lower than those of Flows 4 and 5 (see Fig. 3).

The offline algorithm further tunes the token rate of Flows 2 and 3 to maximize the throughput, so that the flows which shares the common bottleneck link (C2-E4) achieve similar throughput (see Fig. 4). The end-to-end throughput of Flows 2 and 3 is 20% increased when the offline algorithm is applied. The recalculated token rate allows Flows 2 and 3 to utilize the unused bandwidth of link (C1-C2), even if the loss probability (contention loss) could be increased a bit more than that of the online algorithm.

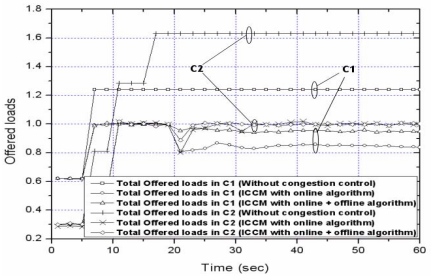

Fig.2.Total offered load in the network.

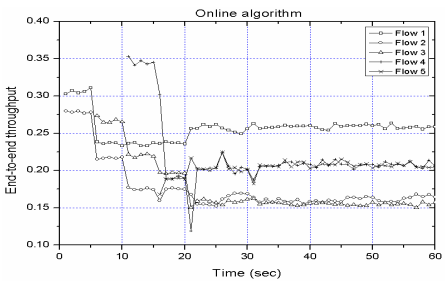

Fig.3.Throughput of online algorithm

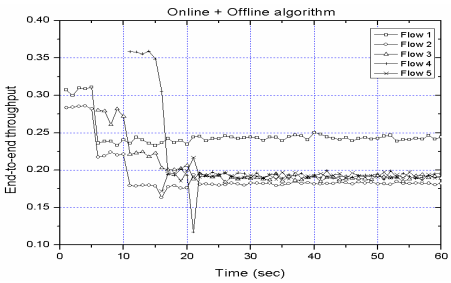

Fig.4.Throughput of online+offline algorithm

Also, we can see that all flows converge to the optimal transmission rate very fast without any oscillations. As a result, the offline algorithm fully utilizes the available bandwidth of each link as well as guarantees an optimized performance.

C. Distance Fairness

In an OBS network, the flows which are transmitted over a multi-hop path experience higher loss probability than single-hop flows, since the loss occurs at the channel-scheduling time, resulting in reduced throughput for a longer path. Our next simulation experiment considers distance fairness, which guarantees the fairness between different hop-count flows.

Figure 5 shows a network scenario with 4 flows for distance fairness. All sources (E1, E4, E5, E6) have enough data bursts to satisfy their token rates, and they seek to maximize throughput. In this scenario, congestion occurs at core routers C2, C3, and C4. Flow 1 traverses all three congested routers, but the other three flows traverse only one congested router. The simulation results are shown in Fig. 6 where the distance factor β varies from 0 to 0.3.

We find that ICCM gives unfair bandwidth allocation when the distance factor is equal to 0. However, larger value of β reduces the unfairness, and eventually, when β is 0.3, the unfairness caused by hop count is nearly eliminated because the distance factor β gives priority to multi-hop flows. As β is increased, the multi-hop flows get more priority than single-hop flows. In this scenario, the data bursts which belong to Flow 1 experience low intentional-drop probability compared to Flows 2, 3, and 4 at the core routers, such that the fair bandwidth allocation between different hop-count flows can be achieved.

VI. CONCLUSION

In this paper, we have proposed and investigated the characteristics of a new integrated congestion-control mechanism (ICCM) for OBS networks. ICCM, which relies on both congestion avoidance and recovery in an integrated manner, is able to prevent congestion effectively by using feedback-control-based rate control and a flow-policing mechanism. The feedback-control-based rate-control mechanism, which is responsible for congestion avoidance, ensures that, at the edge router of the network, each flow's bursts do not enter the network at a rate higher than that the network can handle, while the flow-policing scheme ensures congestion recovery in the network by dropping excessive loads, and supports fairness performance between flows. Also, the rate controller which adopts a SA-based optimum rate-decision algorithm as an optimization tool, allows optimized performance of each flow. Simulation results show that ICCM successfully prevents overloads while it is able to achieve fairness and optimized performance for competing network flows.

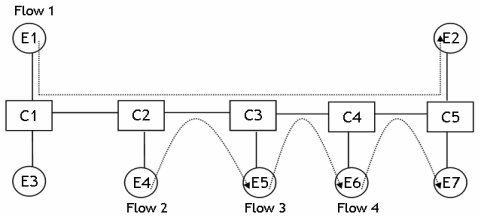

Fig. 5. Network scenario for distance fairness.

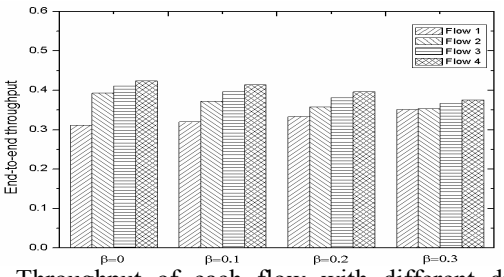

Fig. 6. Throughput of each flow with different distance factor.

ACKNOWLEDGMENT

This research has been conducted while Sungchang Kim was visiting the Networks Research Lab. at UC Davis. This work was supported by KOSEF through OIRC project, and also in part by NSF Grant No. INT-03-23384.

REFERENCES

[1] S. Kim, N. Kim, and M Kang, "Contention Resolution for Optical Burst Switching Networks Using Alternative Routing," *Proc., ICC 2002,* vol. 5, pp. 2678–2681, May 2002.

[2] V. Vokkarane and J. Jue, "Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks," *IEEE JSAC*, vol. 21, no.7, pp. 1198–1209, Sept. 2003.

[3] F. Farahmand and J. Jue, "A Feedback-Based Contention Avoidance Mechanism for Optical Burst Switching Networks," *OBS Workshop at BroadNets04*, Oct. 2004.

[4] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Control Architecture for Optical Burst Switched WDM Networks," *IEEE J SAC*, vol. 18, no.10, pp. 1838–1851, Oct. 2000.

[5] L. Kleinrock, *Queueing Systems, Volume 1: Theory*, New York: Wiley Interscience, 1975.