EE384Y Project Final Report

Load Balancing and Switch Scheduling

Xiangheng Liu
Department of Electrical Engineering
Stanford University, Stanford CA 94305
Email: liuxh@systems.stanford.edu

Abstract

Load balancing and switch scheduling are two important algorithms in the effort to maximize the stability region and minimize (average) latency. Load balancing regulates the traffic to conform to the service rate while the switch scheduling allocates the service rates adaptive to the arrival patterns. Many existing load balancing and switch scheduling algorithms share great resemblance. We show that the load balancing and switch scheduling problems are dual systems of each other based on the linear queue dynamics approximation. This allows us to cast a load balancing problem in terms of a scheduling problem and vice versa. We further show an example of designing a new algorithm for load balancing using an existing scheduling algorithm based on the duality. We also explore the possibility of finding the entropy rate of the randomized load balancing system based on the duality and current knowledge of the entropy rate of randomized load balancing system [5].

The load balanced switch has stirred a lot of interest for its simplicity and performance. We conjecture the joint use of more sophisticated load balancing and switch scheduling algorithms will further improve the performance. We use mean field analysis to show such performance gain in the one dimensional case.

I. Introduction

Load balancing is a fundamental problem in many practical scenarios. A familiar example is the supermarket model where a central allocator assigns each arriving customer to one of a collection of servers to minimize the expected delay. The intuitively ideal SQ (join the shortest queue) algorithm is optimal but the implementation for large systems can be costly. Various randomization algorithms were proposed [2] to reduce the complexity of the algorithm while keeping the good performance. The use of memory in randomized load balancing has been proven attractive. It was shown in [3] that memory gives a multiplicative effect instead of an additive effect for performance improvement.

Switch scheduling determines which inputs to connect with which outputs in every time slot. It is well known that the crossbar constraint makes the switch scheduling problem a matching problem in a bipartite graph [4]. Even though the scheduling problem appears to be solved by completely different techniques from load balancing, we observe that many scheduling algorithms have a counterpart in load balancing algorithms. For example, SQ vs. LQF, RAND, RAND(d) and randomized algorithms with memory RAND(d,m), etc. In this project, we aim for a fundamental relationship in finding load balancing and switch scheduling algorithms. We show that the two problems can be cast into each other using a negative dual transformation. We can use this duality to come up with new algorithms and solve new problems.

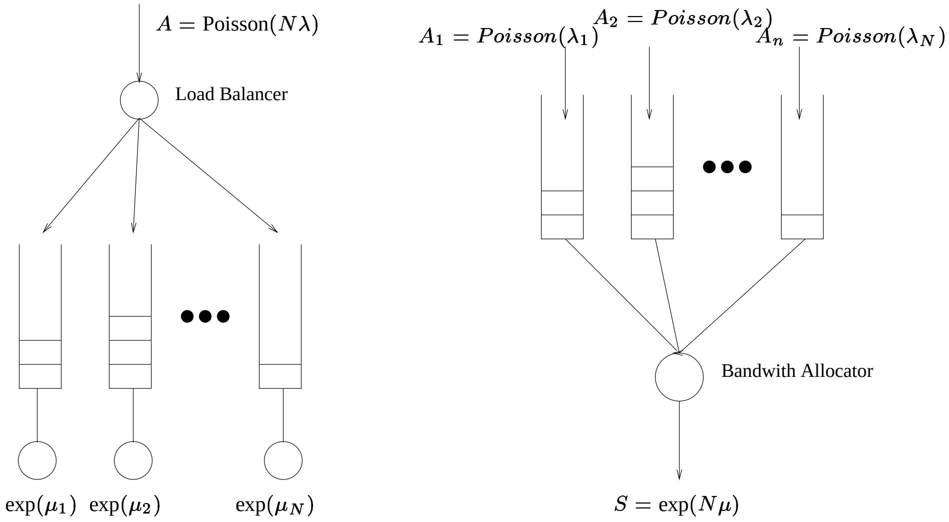
$A = \text{Poisson}(N\lambda)$
Load Balancer
$\exp(\mu_1)$
$\exp(\mu_2)$
$\exp(\mu_N)$
$A_2 = Poisson(\lambda_2)$
$A_1 = Poisson(\lambda_1)$
$A_n = Poisson(\lambda_N)$
Bandwith Allocator
$S = \exp(N\mu)$

Fig. 1. (a) One Dimensional Load Balancing (b) Bandwidth Allocator

We also observe that dual algorithms usually work well together. This is often due to the mathematical duality that is fundamental to the system. We study the joint system with dual load balancing and switch scheduling algorithms for the one-dimensional system. We show the performance gain using mean field analysis. The two-stage load balanced Birkhoff-von Neumann switches discussed in [1] involves load balancing in the first stage where load balancers allocate all traffic uniformly across all virtual output queues with the correct destination. We conjecture that the joint use of more sophisticated load balancing and scheduling algorithms give much better performance.

The rest of the paper is organized as follows. We show the equivalence of load balancing and switch scheduling algorithms for one dimensional system in Section II. We extend the argument to an N by N switch in Section III, where we also illustrate how the duality result can be useful. We will discuss our attempt to solve the entropy rate problem for randomized bandwidth allocation based on duality in Section IV. We analyze the performance of jointly using dual load balancing and switch scheduling algorithms for the one-dimensional system in Section V. Finally, we conclude in Section VI.

II. 1-D Scenario: Load Balancing and Bandwidth Allocation

In this section, we consider one dimensional load balancing and switch scheduling algorithms. In a one dimensional load balancing system, a single packet stream arrives at a load balancer and the load balancer allocates each packet to one of the N servers with individual queues. The one dimensional switch scheduling algorithm is often referred as bandwidth allocation. All N input queues share one server and the scheduler determines which queue to serve at each time slot. Figure 1 shows the one dimensional load balancing and scheduling problem.

We assume the arrivals happen at the beginning of a time slot and departures occur at the end of the time slot and the packet buffer length is measured in the middle of the time slot. Let $q_i(n)$ denote the i^{th}

queue length at time slot n and $A_i(n)$ and $D_i(n)$ denote the number of arrivals and departures in time slot n, respectively. Note $A_i(n)$ and $D_i(n)$ only take binary values. For any queue i, we have

$$q_i(n+1) = [q_i(n) - D_i(n)]^+ + A_i(n+1). \tag{1}$$

The dynamics of the queue is not linear and it is often approximated as a linear system:

$$q_i(n+1) \approx q_i(n) - D_i(n) + A_i(n+1). \tag{2}$$

In a load balancing system, the load balancer controls the arrival to each queue $A_i(n)$ while $D_i(n)$ is given by the service discipline. On the other hand, switch scheduling algorithm determines the departures from each queue $D_i(n)$ while $A_i(n)$ is not controllable.

Note the approximate system dynamics have an equivalent representation:

$$-q_i(n+1) \approx -q_i(n) + D_i(n) - A_i(n+1). \tag{3}$$

If we set $\hat{q}_i(n) = -q_i(n)$, $\hat{A}_i(n+1) = D_i(n)$ and $\hat{D}_i(n) = A(n+1)$, then we have

$$\hat{q}_i(n+1) \approx \hat{q}_i(n) + \hat{A}_i(n+1) - \hat{D}_i(n). \tag{4}$$

Note this is exactly the same as Equation 2. If we start with a bandwidth allocation problem that has system dynamics as in 2 and we need to determine $D_i(n)$, we arrive at a load balancing problem as in 4 and we need to determine $\hat{A}_i(n+1)$, the equivalent of $D_i(n)$. This equivalence is achieved by considering a negative dual system. By negative, we mean that the new system has a state variable that is the negative of the original system. By dual system, we mean the arrival and departure processes are swapped. Essentially, one can imagine this as reversing all the arrows in Figure 1(b) and consider the bandwidth allocator as a token allocator. If a token arrives at queue i, then one packet can leave that queue. Hence, the number of packets waiting in the queue is equal to the negative of the number of tokens in the queue. By doing this, we can cast the bandwidth allocation problem as a load balancing problem. Note this is only true for the approximate queue dynamics. We would need to verify how well this approximation works in the simulations.

Another technical point is that the time indexes change when we swap the arrivals and departures. This is due to our convention that arrivals occur at the beginning of the time slot and departure occur at the end of the time slot. Between the measurement of $q_i(n)$ and $q_i(n+1)$, there is one arrival $A_i(n+1)$ and one departure $D_i(n)$. Thus we swap the arrival and the departure within the time of measuring $q_i(n)$ and $q_i(n+1)$, we need to change their time indexes as well to conform with our assumption.

By considering the negative dual system, we arrive at two identical systems except for the sign of the state of the system. Hence, the load balancing algorithm SQ (join the shortest queue) will lead to the bandwidth allocation algorithm LQF (Longest Queue First). Also due to the equivalence in the system dynamics, when a version of RAND, RAND(d) and RAND(d,m) is used in load balancing, the performance of the switch is similar to what can be achieved by the dual algorithm in switch scheduling.

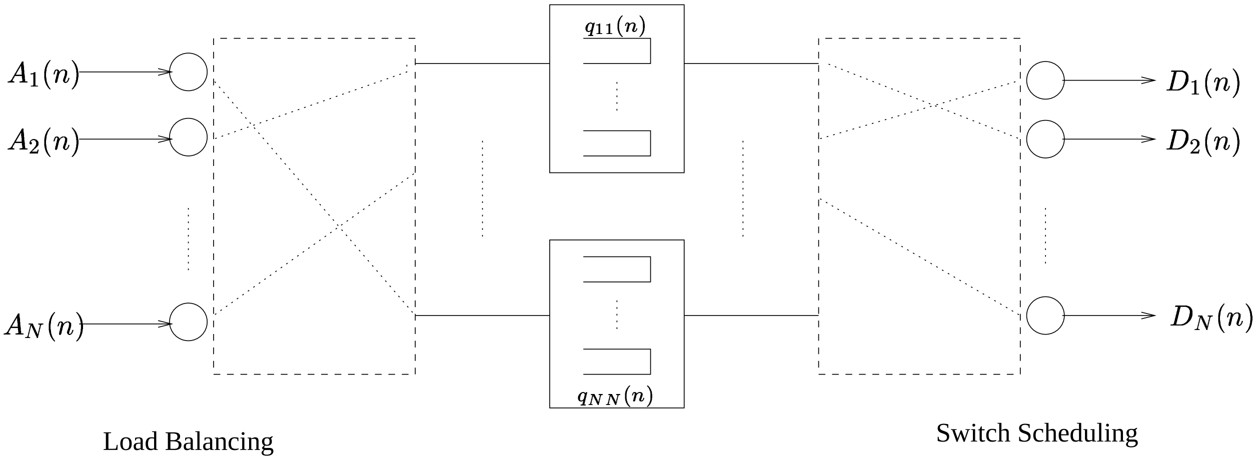

Fig. 2. Load Balancing and Switch Scheduling in Two-stage Load Balanced Switch

Since the queue length process is a Markov chain, the time reversibility is often studied. Basically, $q_i(n+1) \approx q_i(n) - D_i(n) + A_i(n+1)$, is equivalent to $q_i(n) \approx q_k(n+1) + D_i(n) - A_i(n+1)$. If we swap the arrivals and the departures, the queue dynamics also become the same. This may imply that the load balancing in the forward time is equivalent to the bandwidth allocation in the reverse time. However, we are not able to explain some of the resemblance with this equivalence. For example, the SQ policy in load balancing should stay optimal for the the bandwidth allocation, which is not true.

III. 2-D Scenario: Crossbar Load Balancing and Switch Scheduling

The dual relationship in Section II can be extended to the two dimensional scenario as in a common NxN switch. A packet that arrives at an input i has a certain destination j. The load balancing refers to the allocation of a packet to input i with destination j to one of the VOQs of q_{kj} for any $k = 1, 2, \cdots, N$. Thus we have N load balancers at the input side. These load balancers are coordinated by a single load balancing algorithm. Note at any given time slot, we can only have one packet arriving at a given input. We assume zero queuing at the load balancers. Hence, all the packets arrive at time slot n need to be transported to one of the VOQs in the same time slot. We also constrain the N parallel load balancers so that only one of the load balancer can allocate its packet to any VOQ's q_{ij} with the same i. Basically, we only allow one read operation at any given input at any time. These assumptions ensure the load balancing problem is essentially a matching problem. The scheduling algorithm is well studied in the NxN switches. It has been shown that the scheduling is the same as a bipartite graph matching and many matching algorithms have been designed. The joint use of the load balancing and switch scheduling forms a two-stage load-balanced switch similar to the one proposed in [1] except we propose to use more sophisticated load balancing and scheduling algorithms.

Similar to the one dimensional case, the switch scheduling algorithm can be cast as a load balancing problem if we consider the negative queuing system with departures and arrivals swapped. This time, we cast load balancing as a switch scheduling problem since the scheduling is much better studied for the crossbar switch. However, there is some technicality. In a switch scheduling algorithm, an edge can be useless if there are no packets from its input port to its output port. However, the matching algorithms deal with this problem naturally. For load balancing, an edge is useless when there is no arrival to that input port at a given time slot. But this is not usually not taken care of in the matching algorithms. This can be easily fixed. At time n, if there is no packet arriving at input port i, we assume $q_{ij} = \infty$ for all $j = 1, \cdots, N$. Thus we can apply the MWM (Maximum Weight Matching) (and potentially many other scheduling algorithms) to load balancing and get a new load balancing algorithm. The counterpart of MWM is essentially a minimum weight matching with the appropriate queue lengths set to ∞ if no packets arrive in the current time slot.

IV. Entropy Rate of Load Balancing and Switch Scheduling

In Section II and III, we have shown that load balancing and switch scheduling are dual systems of each other for the linear dynamics approximation. If the linear approximation proves itself to be an accurate one, we would expect the property of one system holds true for its dual system. In this section, we study the entropy rate of the randomized scheduling based on the knowledge of the entropy rate of randomized load balancing as studied in [5].

The one dimensional load balancing system as shown in Figure 1(a) is studied in [5]. In particular, it is assumed that all servers have identical service rates $\mu_i = 1$ for $i = 1, \cdots, N$. The class of the randomized load balancing algorithms that can be described by a coin toss model is considered. We review the coin toss model defined in [5] as follows. Let $\sigma(k)$ be the permutation of numbers of $1, 2, \cdots, N$ which arranges the queues in the increasing order in time slot $k-1$ right after departures $D_i(k)$. Let $p = (p_1, \cdots, p_N)$ be a probability vector representing the probabilities of the outcome of the toss of a coin with N sides and let $p_1 \geq p_2 \geq \cdots \geq p_N$. If a packet arrives in time slot k, we toss an N-sided coin distributed according to p. If the outcome of the coin toss is C, $1 \leq C \leq N$, then the packet joins the queue $\sigma_C(k)$. The randomized algorithm SQ(d) can be identified with

$$p_i = \frac{\left(\begin{array}{c} N-i+1 \\ d \end{array}\right) - \left(\begin{array}{c} N-i \\ d \end{array}\right)}{\left(\begin{array}{c} N \\ d \end{array}\right)}. \tag{5}$$

The special case of $d = 1$ can be identified with the vector $(\frac{1}{N}, \cdots, \frac{1}{N})$.

The main theorem gives a closed-form formula for the entropy rate of the load balancing algorithms that can be specified by the coin toss model.

Theorem 1: Suppose the arrival process to the N-queue system is stationary, ergodic and renewal. Let the service distribution be independent of the arrival process and i.i.d. Under mild technical conditions [5], the entropy rate of the queue-size process of any algorithm that belongs to the coin toss model is equal to $\lambda(H_{ER}(A)+H(S)+H(C))$, where A,S,C are random variables representing the inter-arrival time, the service time and the coin toss result respectively.

The problem of the entropy rate of the bandwidth allocation problem shown in Figure 1 (b) can be similarly described as a coin toss model. We let $\sigma(k)$ be the permutation of $1,\cdots,N$ and it arranges the queue sizes in decreasing order in time slot k right after arrivals $A_i(k)$. The coin toss probability for LQF(d)[1] is the same as that of SQ(d). We toss a coin to decide which queue to serve.

When all the queues are always non-empty, the linear queue dynamics are exact. Hence, the duality holds. We could use the negative dual transform shown in Section II and cast the bandwidth allocation problem in the language of load balancing. Let $\hat{q}_i(k)=-q_i(k), \hat{A}_i(k+1)=D_i(k), \hat{D}_i(k)=A_i(k+1)$ for $i=1,\cdots,N$, then the randomized bandwidth allocation problem is precisely a load balancing problem that can be described as follows. Let $\sigma(k)$ be the permutation of $1,\cdots,N$ and it arranges the $\hat{q}_1(k),\hat{q}_2(k),\cdots,\hat{q}_N(k)$ in the increasing order in time slot $k-1$ right after departures $\hat{D}_i(k-1)$. Then we do the coin toss. Note this description is exactly the same as the coin toss model for the load balancing algorithm. We can also swap the arrival process and departure process since they have exactly the same properties. For example, with Poisson arrivals and exponential services, both inter-arrival and inter-departure times are exponential. Therefore, exchanging the arrivals and departures will not lead to inconsistency problems. Hence, we claim the entropy rate of the two systems are equal. This leads us to the following proposition.

Proposition 1: If all the queues in the system are always non-empty, the entropy rate of the queue dynamics is the same for the load balancing and switch scheduling systems with correspondent parameters. Hence, the entropy rate of the load balancing system with i.i.d. arrivals of rate λ for each queue has the entropy rate of $\lambda(H_{ER}(A)+H(S)+H(C))$.

However, Proposition 1 does not hold when the queues can be empty since the queue dynamics is no longer linear. We believe this is due to the loss of inter-changeability of the inter-arrival and inter-service times. Let us assume Poisson arrivals and exponential services. For load balancing and bandwidth allocation systems, the inter-arrival times are exponential, but the inter-service times are exponential plus some potential idle time between services due to empty queues. However, when we consider the negative dual system of the bandwidth allocation, the new arrival process has an inter-arrival time of the sum of exponential random variable plus some random idle time. Thus, we no longer have the exact mapping. We denote the random time between serving the k^{th} packet and the $(k+1)^{st}$ packet as $I(k)$. Note $I(k)$ is always equal to zero when all the queues are always non-empty. The bijection that proves Theorem 1 no longer holds. However, we have two injections that give a lower bound and an upper bound on the entropy rate.

Proposition 2: The entropy rate of the bandwidth allocation system is upper bounded by $\lambda(H_{ER}(A)+H(S)+H(I)+H(C))$ and lower bounded by $\lambda(H_{ER}(A)+H(S+I))$.

[1] In LQF(d), we randomly choose d samples in each time slot and serve the longest queue among the d samples. If all the d queues are empty, no packet will be served in the current time slot.

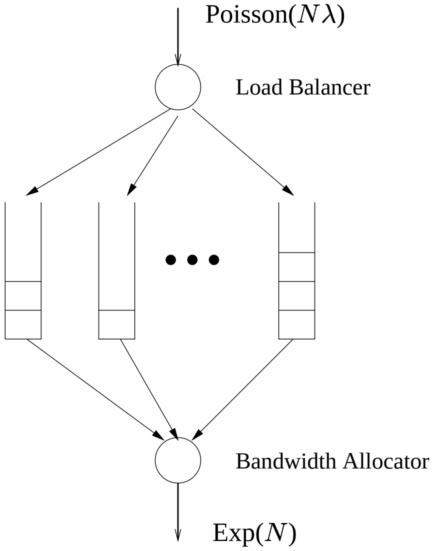

Fig. 3. Joint Load Balancing and Bandwidth Allocation in 1-D system

Outline of the Proof: We follow all the notations in [5]. In addition, we define VI_{k+} be the future inter service idle times for the packets in the queue at time $k+$. We also assume VI_{k+} has finite entropy for any k. We have the following injections:

$$(Q_0, V_{0+}, VI_{0+}, a_1, A^{N(k)-1}, S^{N(k)}, I^{N(k)}, p^{N(k)}) \rightarrow (Q_0, \cdots, Q_K).$$

This gives the upper bound. We also have

$$(Q_0, \cdots, Q_K) \rightarrow (a_1, A^{N(k)-1}, (S+I)^{N(k)}).$$

This gives the lower bound.

However, these bounds are not tight. The search for tighter bounds will be future research.

V. Mean Field Analysis for Joint Load Balancing and Bandwidth Allocation

One of the original motivations for this project is to design a load-balanced switch where the first stage employs a load balancing algorithm and the second stage uses the dual switch scheduling algorithm. The motivation of using dual algorithms come from the observation that the joint use of dual algorithms often achieve optimal performance in linear systems, for example, the Kalman filter followed by a state feedback controller is optimal for LQG (Linear Quadratic Gaussian) control. We believe such joint system gives better performance gain than the systems where only either load balancing or switch scheduling algorithm is adopted. We show the performance gain in the one dimensional system as in Figure 3. We extend the mean field analysis to the joint load balancing and scheduling system.

We consider a system where the arrival is Poisson($N\lambda$) and a load balancer allocates all the packets to a bank of N queues. We assume that these N queues share the same server that operates at rate N.

A bandwidth allocator at the server side determines which queue it serves when the server is free. We compare the joint load balancing and switch scheduling system with the system where only load balancing algorithm is used with the simple random scheduling algorithm.

For load balancing algorithms, we know that the SQ is the optimal but the complexity is prohibitive when N is large. In order to trade for complexity, randomized algorithms are proposed. RAND is the most simple and it delivers the arriving packet to a random chosen queue with equal probabilities. With RAND, the system is the same as N independent $M/M/1$ queues. SQ(d) is a good compromise of the SQ and RAND. SQ(d) picks d random samples and allocate the packet to the shortest queue among the d samples.

The mean field analysis of SQ(d) is discussed in [6] and it was shown that the cumulative distribution of queue length is $P(Q_1 \geq i) = \lambda^{\frac{d^i-1}{d-1}}$. Now let us consider the system where SQ(d) is used for load balancing while LQF(d) is used for the bandwidth allocation. Let $s_i(t)$ denote the fraction of the queues with load at least i at time t. Then $s_i(t)$ satisfy the following set of differential equations.

$$\frac{ds_i(t)}{dt} = \lambda(s_{i-1}^d(t) - s_i^d(t) - 1((1 - s_{i+1}(t))^d - (1 - s_i(t))^d). \tag{6}$$

In equilibrium, $\frac{ds_i(t)}{dt} = 0$. Since the above equation is true for all i,

$$\sum_{k \geq i} \lambda(s_{k-1}^d(t) - s_k^d(t) = \sum_{k \geq i}(1 - s_{k+1}(t))^d - (1 - s_k(t))^d. \tag{7}$$

This gives

$$s_i(t) = 1 - (1 - \lambda s_{i-1}^d(t))^{\frac{1}{d}}. \tag{8}$$

By the law of large numbers, $P(Q_1 \geq i) = s_i(t)$. Since $s_0(t) = 1$ for all t. We can find the distribution of the queue length using recursions. We compare the joint $SQ(2)$ and $LQF(2)$ with $SQ(d)$ only and $LQF(d)$ only $d = 2, 3$. The figure 4 shows the joint use of $SQ(2)$ and $LQF(2)$ performs much better than $SQ(2)$ along or $LQF(2)$ alone.

VI. Conclusions

We study the similarity between load balancing and switch scheduling algorithms. We show that the two problems are equivalent based on a negative dual transformation for the linear queue dynamics approximation. This duality can help us come up with new algorithms in load balancing based on the existing scheduling algorithms and vice versa. The duality also directly leads to the entropy rate of the bandwidth allocation system when the linear queue dynamics are exact (all the queues in the system are always non-empty) since we already know the entropy rate of the load balancing system. However, when the the queues do not have linear dynamics, we are not able to find the exact entropy rate, instead, we find an upper bound and a lower bound. We are interested to search for tighter bounds in future research.

The joint use of load balancing and switch scheduling are shown to improve the performance in the one dimensional system. We conjecture that similar performance gains can be obtained for the two dimensional system as well.

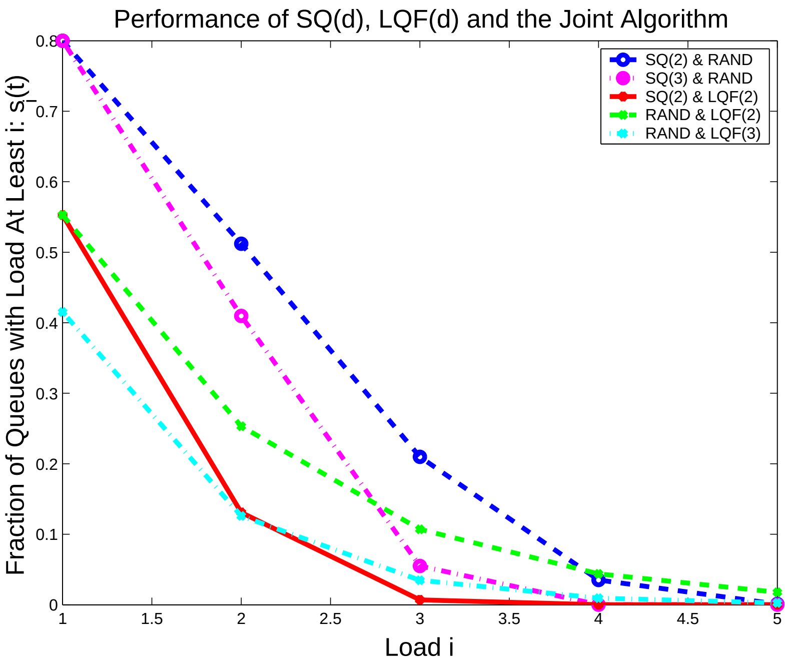

Fig. 4. Performance of Joint Load Balancing and Bandwidth Allocation

Acknowledgments

The authors would like to thank Isaac Keslassy for his helpful discussions.

References

[1] C.S. Chang, D.S. Lee, Y.S Jou, “Load balanced Birkhoff-von Neumann Switches” *IEEE Workshop on High Performance Switching and Routing, 2001.*

[2] A. Czumaj and V. Stemann, “Randomized Allocation Processes”, *FOCS, 1997.*

[3] Devavrat Shah, Balaji Prabhakar, “The use of memory in randomized load balancing”, *ISIT 2002.*

[4] P. Giaccone, B. Prabhakar, D. Shah, “Randomized Scheduling Algorithms for High Aggregate Bandwidth switches”, *Infocom 2002.*

[5] C. Nair, B. Prabhakar, D. Shah, “The Randomness in Randomized Load Balancing”, *Proceedings of the 39th Annual Allerton Conference on Communication, Control and Computing*, pp.912-921, October 2001.

[6] N. McKeown, B. Prabhakar, *EE384Y Lecture Notes*, Stanford Unviersity 2003.