

Dynamic Scheduling in Queueing Systems with Applications to Communication Networks

Kevin Ross
Ph.D. Thesis
Management Science and Engineering
Stanford University, 2004

Abstract

This thesis develops online scheduling algorithms for important applications arising in computer and communication networks. These are modeled through a general framework of service allocation over a multiplicity of interdependent queues. The work is motivated by large scale data scheduling in high performance packet switches and burst scheduling in an optical fibre network, with further implications for call centers and flexible manufacturing.

The throughput capacity of the appropriate queueing system model is established under minimal arrival process restrictions, and classes of allocation algorithms are proposed to maximize throughput and optimize quality of service/load balancing criteria. Given the computational complexity arising from the sometimes prohibitively large number of configurations and queues, it is also shown how these algorithms can be implemented using a local search. Robustness issues including incomplete, delayed and noisy information are also considered.

The model considers a generalized packet switch as a processing system having several queues, where $X_q(t)$ is the backlog (number of cells) in queue q at time slot t . In each time slot, the system can be set to a single service configuration S , chosen from the set \mathcal{S} of all feasible ones. When the system is set to S in a time slot, S_q cells are removed from queue q in that slot. The objective is to dynamically choose and schedule the service configurations in consecutive time slots, so as to maximize the system throughput. By considering *vectors* $X(t)$ and S , the rich geometry of system dynamics is highlighted, providing intuition into the stability analysis techniques and parameter selection for load balancing.

In the special case of a crossbar packet switch, each queue stores packets waiting to be sent between a particular input and output port pair. A packet arriving to the input port is

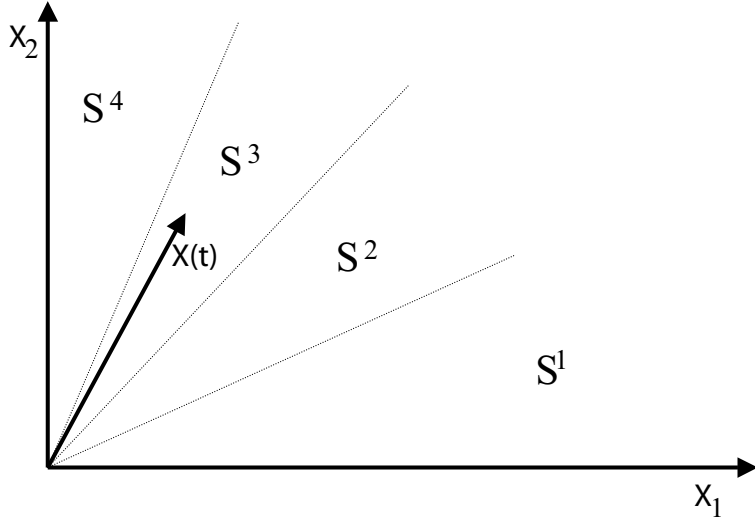


Figure 1: **Geometric Workload Evolution.**

The backlog state $X(t)$ is a vector in Q dimensions. The backlog-based service policies select the service configuration based on the backlog state. For PCS, this means selecting the service rates according to a cone policy; when $X(t)$ is in the cone of configuration S , select that configuration.

stored in this virtual output queue until the switch is configured to connect it to its output port. Since each port can establish exactly one connection in each timeslot, the set of available configurations corresponds to the matchings of input and output pairs. In particular, service vectors have only $(0,1)$ entries, as opposed to the more general case where an arbitrary number of cells can be processed in each timeslot.

Throughput maximization for this generalized switch system is achieved by a broad family of *projective cone scheduling (PCS) algorithms*, which choose a service vector $S \in \mathcal{S}$ that maximizes the inner product

$$\langle S, \mathbf{B}X \rangle = \sum_p \sum_q S_p B_{pq} X_q \quad (0.1)$$

when the backlog vector is X , for any fixed matrix $\mathbf{B} = \{B_{pq}\}$ which is *positive-definite*, *symmetric*, and has *negative or zero off-diagonal elements*. These schedules substantially generalize the Maximum Weight Matching (MWM) algorithms, where \mathbf{B} is simply the identity matrix and \mathcal{S} is the set of matchings in a crossbar switch.

It has been observed that weighting-based scheduling policies such as PCS algorithms can be very computationally intensive and potentially impractical in some switching application

scenarios, because the set \mathcal{S} may have a huge number of elements S to calculate the inner product at. Therefore, a new class of PCS-based algorithms is developed using ‘local search’ concepts. In particular, rather than searching the entire (typically huge) set of available service configurations to find the best one, the new scalable scheduling algorithms search ‘locally’ over a small neighborhood of service configurations to find a ‘better’ one in each time slot. This analysis shows that local projective scheduling algorithms can provide dramatic reduction in complexity *without* causing any loss of throughput (although they may observe higher delay). This thesis explores the nature and structure of such schedules, which show a much higher promise for practical implementation than their global versions.

An important extension highlighted in this work is the case where propagation delays in a network cause contention over multiple time periods. For example, this occurs in a Time-domain Wavelength Interleaved Network (TWIN), an optical network with a single tunable laser and receiver at each node. Due to the high data rates employed on optical links, the burst transmissions typically last for very short times compared to the round trip propagation times between source-destination pairs. An intelligent scheduling algorithm must take these propagation delays into account in order to avoid conflicts and maximize the throughput of the network. For such applications with propagation or switching delays, adaptive batch scheduling (ABS) algorithms are introduced, which schedule sequential batches of service. For the TWIN application, it is shown that any greedy algorithm performs with an inefficiency factor no larger than 2 on arbitrary instances of this problem, and an improved static scheduling algorithm is proposed and shown to have near optimal performance in many examples.

This research develops a novel but intuitive trace-based approach to performance modeling and analysis. Arrivals can follow arbitrary dynamics, and are assumed only to have a (perhaps unknown) long-term arrival rate to each queue. In particular, this allows for arrivals that are correlated across multiple queues or follow any statistical patterns. Even under such minimal assumptions, both PCS and ABS algorithm classes are proved to guarantee maximal throughput, and it is shown how to tune the appropriate parameters in order to meet load balancing and complexity reduction objectives.