An Overview of The Application of Heavy Traffic Theory and Brownian Approximations to the Control of Multiclass Queuing Networks

Elif Uysal

1 Introduction

Stochastic networks are used to model communication networks as well as other networks such as those in complex manufacturing systems. These networks are characterized by having entities such as packets, jobs, customers (in this report we will call these jobs), that receive processing at various servers (resources), wait in buffers (queues), and which, in general, may have random arrival times, random processing times, and routing protocols. These networks often cannot be analyzed exactly. Among approximation methods used for the analysis and control of such systems, this week and next week we will look at two levels of approximation: first order (functional law of large numbers) approximations called fluid models (which will be one of the topics of next week), and second order (functional central limit theorem) approximations which are diffusion models (which will be the focus of this summary.)

The term “heavy traffic” is used to mean that the nominal load on the system is approximately equal to the system's capacity. In this regime, it is sometimes possible to approximate the stochastic processes of interest by processes which are easier to analyze, such as Brownian motion. Heavy traffic analysis can provide conditions under which such approximations are mathematically justified. This area of research has a strong tradition going back to the 1960's and has reached a considerable level of mathematical maturity.

An important class of stochastic processing networks is open multiclass queueing networks operating under a head-of-the-line (HL) service policy. In a multiclass queueing network, there is a many-to-one function describing the association between buffers and resources (also called servers), and a HL policy for such a network assumes that jobs in a queue are ordered and selected for processing by the associated resource on a first-in-first-out (FIFO) basis. Since the late 1980's an extensive mathematical theory has been developed [*e.g.*, Riemann, Harrison, Williams, Bramson] for using fluid and diffusion approximations to analyze the stability and heavy traffic performance of open multiclass HL queueing networks. The goal of this summary will be to provide a simple introduction and to try to illustrate some important ideas and results such as convergence to a **reflected Brownian motion** under **heavy traffic scaling**, **resource pooling**, and **state-space collapse**. We will introduce those notions inside scheduling paradigms such as minimum-cost scheduling in a multiclass queuing system, staying mainly within the frameworks of [Harrison and Lopez 1999], [Harrison and Van Mieghem 1997] and [Harrison 1988].

2 Preliminaries

For the definition of Brownian motion, we quote [Harrison 1985]: A stochastic process X(t) is said to have independent increments if the random variables $X(t_1) - X(t_0)$, ..., $X(t_n) - X(t_{n-1})$ are independent for any $n \geq 1$ and $0 \leq t_1 < t_2 < \ldots < t_n < \infty$. It is said to have stationary independent increments if moreover the distribution of $X(t) - X(s)$ depends only on $t - s$. We will use the standard notation $Z \sim N(\mu, \sigma^2)$ to mean that the random variable Z is Normal (Gaussian) with mean μ and variance σ^2. A $s$tandard Brownian motion, or Wiener process, is defined as a stochastic process $B(t)$ having continuous sample paths, stationary independent increments, and $B(t) \sim N(0, t)$. A Brownian motion Y(t) with $d$rift μ and variance σ^2 has the form $Y(t) = Y(0) + \mu t + \sigma B(t)$. An m-dimensional Brownian motion is defined with a mean vector and a covariance matrix, in the obvious way.

3 The general procedure of heavy traffic analysis

Early work in the development of heavy traffic theory for complex queuing systems include [Reimann 1984], [Harrison 1988] and [Harrison and Williams 1992].

The general procedure laid out in [Harrison 1888] has been applied successfully in many contexts. First, one derives a limiting Brownian control problem that plausibly approximates the original dynamic scheduling problem (after some scaling.) The second step is to solve the Brownian control problem. Because fine structure is supressed in the Brownian problem, it is usually a much simpler problem than the original. Third, using some creativity, one interprets the solution in the context of the original problem. Finally, ideally one would like to prove that the proposed solution is asymptotically optimal in the original problem.

With this general overview of heavy traffic analysis, we are ready to look at the specific problem formulated in [Harrison and Lopez 1999]

4 A parallel-server system with resource pooling [Harrison and Lopez 1999]

Consider the multi-class multi-server queuing system described in Figure 1. The job classes (*i.e.*, the input queues) are indexed by $i = 1, \ldots, m$, and the servers are $k = 1, \ldots, l$. Several different classes may be processed at the same server, several different servers may be capable of processing jobs of the same class. This is summarized by saying that there are $n \leq ml$ activities available to the system manager. Let λ_i be the average arrival rate of class i jobs, and μ_j the reciprocal of the mean service time for activity j. At each queue, there is a "holding cost" that is incurred at a rate of c_i per unit time for each job in the queue. The scheduling problem is that of dynamically allocating jobs to servers so that long-run average holding costs per time unit are minimized.

Note that, since there are multiple queues that can share servers, it is not obvious what is meant by "heavy traffic" in this system. Let ρ be the long-run utilization of the busiest server. Harrison and Lopez start by obtaining the solution of the static allocation problem by solving the following linear program:

$$\begin{array}{rcl} \text{minimize} & \rho & \\ \text{subject to } Rx & = & \lambda \\ Ax & \leq & \rho e \\ x, \rho & \geq & 0 \end{array}$$

In the above, $A_{ij} = 1$ if server i serves activity j, and $A_{ij} = 0$ otherwise. $R_{ij} = \mu_j$ if server i serves activity j, and $R_{ij} = 0$ otherwise. Note that

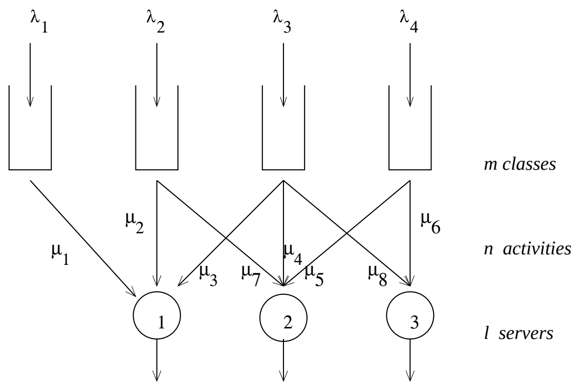

Figure 1: Example for the multiclass parallel server system studied in [Harrison and Lopez 1999]

e is an l-vector of ones. Assuming that the above problem has a unique optimal solution (x^*, ρ^*), the n-vector x^* will be called the *nominal processing plan*, and that x_j gives the long-run proportion of time that activity j is performed by its server. So, the first constraint simply enforces material conservation, the second requires that each server's utilization not exceed the busiest server's. Activities for which $x_j^* > 0$ are called *basic activities*. Let b be the number of basic activities, and assume that the activities $j = 1, \ldots, b$ are the basic ones. Note that when $\rho^* = 1$, the system manager is just able, using the basic activities, to process jobs of the various classes at the required average rates. Any other feasible processing plan would necessarily exceed capacity of at least one server.

The question posed by the authors is "can server work assignments be *dynamically* adjusted, relative to the nominal processing plan x^*, to minimize cost?" Recall that different queues have different holding cost rates in the model. To answer this question, they first make the following definition:

Definition 1. *(Communicating Servers.) Server k communicates* directly *with server k' if there exist basic activities j and j' such that $j = j(i,k)$ and $j' = j(i,k')$ for some class i. Server k* communicates *with server k' if there exist servers $k_1, \ldots, k_w$ such that $k_1 = k$, $k_w = k'$, and k_α communicates directly with $k_{\alpha+1}$ for all $\alpha = 1, \ldots, w-1$.*

Note that communication is an equivalence relation, hence the servers can be partitioned into disjoint sets in which each server communicates only with those servers in its set. There is only one such set in Figure 1. The intuition is that, if all servers communicate, one can schedule jobs (by shifting from one activity to another) such that the most backlog is accumulated at the queue with the smallest holding cost; which would minimize overall cost (Note that this is related to the well-known $c\mu$ rule. This is the idea behind "resource pooling."

To formally introduce resource pooling, Harrison and Lopez introduce the limiting Brownian problem (which, they loosely argue based on earlier theory (see [Harrison 1988], is the heavy-traffic limit of the original problem):

Let $F_i^0(t)$ be the number of arrivals into buffer i in the time interval $[0, t]$. Let $F_i^j(t)$ be the number of departures from buffer i resulting from the first t time units devoted to activity j by its server (note that this is a counting process.) It can be shown ([Reiman 1984], [Harrison 1988]), that the scaled process:

$F^0(rt)/\sqrt{r} - \lambda t\sqrt{r}$

converges weakly as $r \to \infty$ to a Brownian motion with zero limit and with an $m \times m$ covariance matrix Γ^0. [1]

Similarly, for all j and $t \geq 0$

$$F^j(rt)/\sqrt{r} - R^j t\sqrt{r}$$

converge weakly to a Brownian motion with zero drift and covariance matrix Γ^j.[2]

One can describe this scheduling policy by means of an n-dimensional stochastic process $T = \{T(t), t \geq \ 0\}$ with components $T_j(t)$, the total time devoted to activity j by its server over $[0, t]$ (In fact this is quite standard.) The m-dimensional jobcount process is defined as:

$$Q(t) = F^0(t) - \sum_{j=1}^{n} F^j(T_j(t)), t \geq 0$$

The cost rate process is then

$$C(t) = cQ(t)$$

[1] One may wonder here how this functional central limit theorem works, since the sample paths of the processes $F^0(rt)$ for every r have discrete jumps (They are right-continuous with left-limits.) But since the sample paths of Brownian motion are continuous, convergence on the Skorohod J-1 topology is the same as convergence u.o.c. For the definition of weak stochastic convergence, see [Siegman 2002]

[2] Γ^j is the $m \times m$ matrix whose $i(j)$ th diagonal element is $\mu_j^3 \sigma_j^2$ and all other elements are zero.

Now let us state the heavy traffic assumption:

Assumption 1. *(Heavy Traffic)*
The data (R, A, λ) of the static allocation problem are such that

- *its solution (x^*, ρ^*) is unique;*
- *$\rho^* = 1$, and*
- *$Ax^* = e$.*

Under Assumption 1, let us define the centered allocation:

$$V(t) = x^* t - T(t)$$

This is the n-dimensional vector process of deviations from the nominal allocation. The standard "cumulative idleness process" $I(t)$, is defined as the vector of cumulative idle time for the n activities. Obviously, all components of $I(t)$ must be nondecreasing if T is to be an admissible policy. Applying heavy-traffic scaling to these processes, one obtains the scaled processes:

$$\begin{aligned} Y^r(t) &= V(rt)/\sqrt{r} \\ Z^r(t) &= Q(rt)/\sqrt{r} \\ U^r(t) &= I(rt)/\sqrt{r} \text{ and,} \\ \xi^r(t) &= C(rt)/\sqrt{r} \end{aligned}$$

To visualize this scaling, suppose $r = 100$. While $Q(t)$ tells us how many jobs get queued up in the first t seconds, $Z(t)$ talks about how many 10's of jobs in the first $100t$ seconds. We are looking at longer and longer stretches of time, but also scaling the quantities of interest, in a way that should be familiar from elementary central limit theorems. Of course, note that the scaling of the r^{th} system is only appropriate when time spans of order r are relevant for purposes of performance measurement.

5 The limiting Brownian control problem

In [Harrison 1988], it is informally argued that under the heavy traffic assumption, as $r \to \infty$, the scaled dynamic scheduling problem is increasingly well approximated by a simpler dynamic control problem where the relationship between Z and Y is modeled by the linear system equation $Z(t) = X(t) + RY(t)$ involving the m-dimensional Brownian motion $X = \{X(t), t \geq 0\}$. In the present case, X has zero drift, and covariance matrix $\Gamma = \Gamma^0 + \sum_{j=1}^{b} x_j^* \Gamma^j$ and initial state $X(0) = 0$. The limiting Brownian problem then is the following (note that $Y(t)$ is our n-dimensional control):

$$\begin{array}{rcll}
Y & & \text{is non-anticipating with respect to } X & (1) \\
Z(t) & \geq & 0 \text{ for all } \geq 0, \text{ and} & (2) \\
U & & \text{is nondecreasing with } U(0) \geq 0 & (3) \\
Z(t) & = & X(t) + RY(t) \text{ for all } t \geq 0 & (4) \\
U(t) & = & KY(t) \text{ for all } t \geq 0 & (5)
\end{array}$$

In the above problem, Y being non-anticipating simply means that $Y(t)$ uses information only about $X(s), s \leq t$ and not the future of X. In probabilistic terms, Y is adapted to the filtration $\mathcal{F}_\sqcup$, the filtration on which $X(t)$ is defined. A process $P = \{P(t), t \geq 0\}$ is said to be *adapted*if $P(t)$ is measurable with respect to $\mathcal{F}_\sqcup$ for each fixed $t \geq 0$.

At this point, we are ready to see one of the strengths of the Brownian approximation: the Brownian system model above has an "equivalent workload formulation" in which the state of the system at any time t is described by a workload vector $W(t)$ having dimension $d \leq m$. In fact, in the present system, $d = 1$. Specifically, $W(t) = y^* Z(t)$ where y^* is the unique optimal dual solution of the static allocation problem. Under SSC, the simplified problem is to choose the pair (Z, U) such that:

$$\begin{array}{rcll}
U & & \text{is non-anticipating with respect to } X & (6) \\
Z(t) & \geq & 0 \text{ for all } t \geq 0, \text{ and} & (7) \\
U & & \text{is nondecreasing with } U(0) \geq 0 & (8) \\
W(t) & = & \Psi(t) + GU(t) \text{ for all } t \geq 0 & (9) \\
W(t) & = & y^* Z(t) \text{ for all } t \geq 0 & (10)
\end{array}$$

Before considering the solution of this problem, let us digress for a bit to talk about SSC.

6 State Space Collapse

In the problem of Harrison and Lopez, the *state space of the problem has collapsed* from m dimensions to 1 dimension. The state space collapse (SSC) phenomenon was first observed by Whitt [1971]. In the present problem, the reason that the system collapses to 1 dimension is that there is complete resource pooling (CRP). Intuitively, under CRP as in Figure 1, under the heavy traffic scaling, one has enough time to switch all the workload from any queue to any other while keeping the total workload the same. In particular, the system manager can keep as much workload as possible in the class with minimum holding cost c, therefore in the limit only one queue is critically loaded. Note that this is related to the well-known $c\mu$ rule.

Although rigorous treatments of SSC in Brownian systems date back at least to Reiman's 1983 paper, Harrison and Van Mieghem [1997] have a quite approachable treatment, which we will summarize. Consider the Brownian control problem 1 to 5, with an arbitrary intial state z, and suppose that an immediate impulse control $Y(0) = \delta$ is applied at time $t = 0$. Then, the initial values of Z and U will be $Z(0) = z + \delta$ and $U(0) = u$, where $\delta = Ry$ and $u = Ky$. From (3) and (2), one sees that the impulse control is admissible only of $z + \delta \geq 0$ and $u \geq 0$. Let's refer to y as a control increment and to δ as a displacement. If $u = 0$, (that is, $Ky = 0$), then the system manager can immediately apply another control increment of $-y$ which causes a displacement of $-\delta$ and thus returns the system to state z. Thus, if $Ky = 0$, the control increment y is *reversible*. The idea behind SSC is that any two state vectors whose difference is a reversible displacement are equivalent, because a system manager can instantaneously exchange either of those state vectors for the other without affecting the cumulative idleness process U. Hence, in terms of decision making, an adequate summary of the system is given by $W(t) = MZ(t)$, where M is a matrix whose rows are orthogonal to all reversible displacements, meaning $MRy = 0$ if $Ky = 0$. (So, M gets rid of the reversible component of the system state.) It is shown by Harrison and Van Mieghem that $MR = GK$. In the reduced system of d dimensions, the reduced system descriptor W evolves in the absence of control as the $d -$ *dimensional* Brownian motion Ψ, and it depends on the chosen control Y through the process U.

7 Pathwise solution of Brownian control problem

Now, we are ready to talk about the solution of the reduced Brownian problem. Define $L(t) = GU(t)$. Recall from (9) that $W(t) = \Psi(t) + GU(t)$. Let $W^*(t) = \Psi(t) + L^*(t)$ where

$$L^*(t) = -\inf_{0 \le s \le t} \Psi(s), \ t \ge 0$$

Note that $W^*(t)$ is a reflected, or more correctly, regulated Brownian motion with a regulating barrier at 0. Also, note that L^* is continuous and non-decreasing with $L^*(0) = 0$, and by inspection one can see that L^* increases only at times t when $W^*(t) = 0$. From these, it follows that for any admissible strategy (Z, U), the workload process W satisfies $W(t) \ge W^*(t)$ for all $t \ge 0$ (Note that this is a pathwise bound!). It is then easy to show (and we omit the details here) that the cost rate process satisfies, for any admissible strategy, $\xi(t) \ge \xi^*(t)$. Hence the policy (Z^*, U^*) is pathwise optimal. What this policy is doing could be interpreted as follows: the system manager tries to keep the system non-idle. Seeing the workload is approaching zero, rather than idling, it engages some server on a non-basic activity. Since non-basic activities are inefficient, workload rises fast under Brownian scaling, so shortly the system manager can return to a mode in which all servers are fully occupied with basic activities.

The next step is to interpret the policy in the actual queuing system, and propose an algorithm that is asymptotically optimal, or reaches the Brownian solution in the heavy traffic limit.

8 Back to the original queuing system

The authors conjecture that ideal system behavior can be approached through a family of simple scheduling policies related to the $c\mu$ rule. That is, rank classes in increasing order of the index $c_i\mu_i = c_i/y_i^*$, and higher ranked classes are given priority over lower rank ones. They also indicate that there is a simple "discrete review policy" that should be asymptotically optimal.

The simpler problem with multiple classes and a single server, with equal holding costs across classes, was analyzed in [Harrison 1988]. There, it is argued that the server should serve last the class k for which μ_k is smallest, and that this class gets service only when the system is empty of all other classes Note that in the Brownian limit this corresponds to only one queue being in heavy traffic, and a SSC to 1 dimension.

References

[Whitt 1971] W. Whitt, 1971

[Harrison 1985] J. M. Harrison, *B*rownian Motion and Stochastic Flow Systems, John Wiley and Sons, NY, 1985.

[Harrison 1988] J. M. Harrison, Brownian Models of Queuing Networks with Heterogeneous Customer Populations, *S*tochastic Differential Systems, Stochastic Control Theory and Application, 1988, iMA Volumes in Mathematics and Its Applications.

[Harrison and Van Mieghem 1997] The Annals of Applied Probability. 7(3) 747-771, 1997.

[Harrison and Lopez 1999] J. M. Harrison and M. Lopez, “Heavy-Traffic Resource Pooling in Parallel-Server Systems,” Queuing Systems, vol. 33, 1999, 339-368.