Delay Analysis of Switches in Heavy Traffic

Shashibhushan Borade

6.454 - Area I Graduate Seminar

December 8 , 2004

Plan of Action

Heavy traffic scaling: some basics

- Origin of Brownian motion
- Idea of State-space collapse

Switches and Maximum weight matching algorithms

- Stability analysis using fluid scaling
- Characterizing steady state under fluid scaling
- Delay analysis using heavy traffic scaling
  - Only one (input/output) port in heavy traffic [Stolyar'04]
  - All ports in heavy traffic [Shah'04]

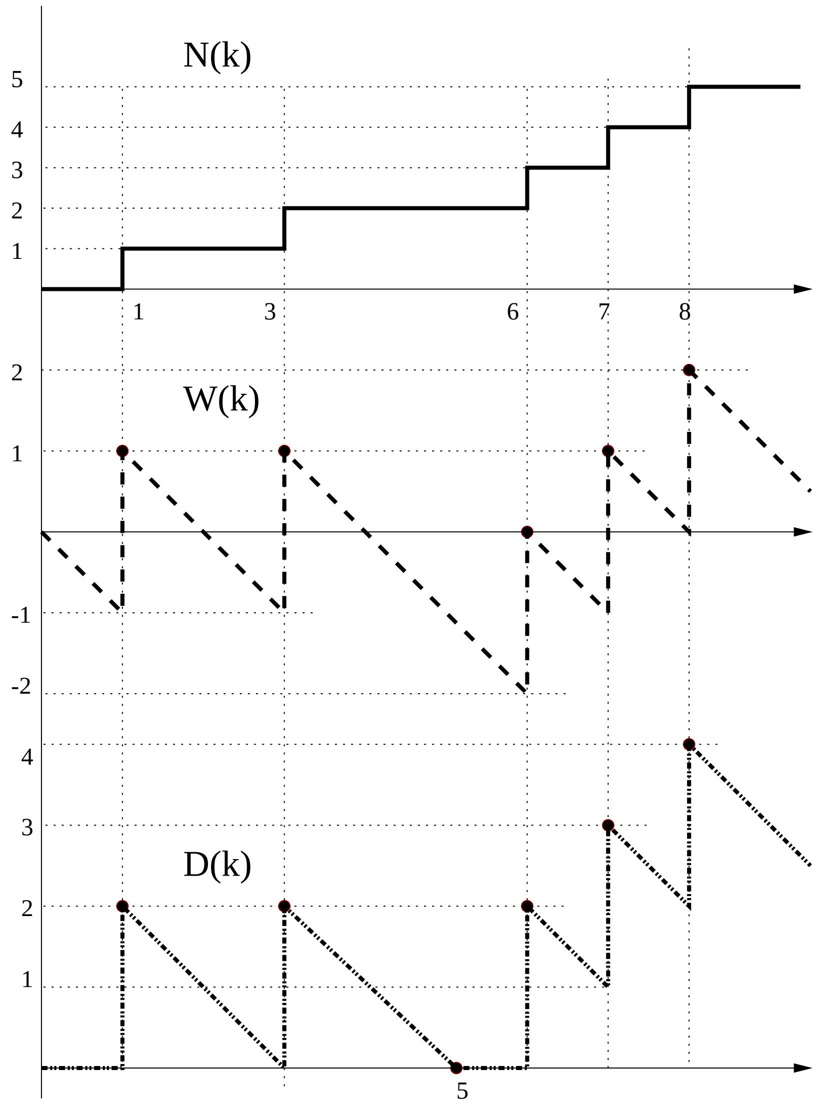
Brownian motion

Single-queue single-server system

Notation

- Arrivals form a renewal process of rate λ
- Inter-arrival times $\{A_i\}$ have mean $1/\lambda$, variance a^2
- All packets have same size $1/\mu$
- – Sum arrivals up to time i: $N(i)$
  – Total time for k arrivals: $T(k) = \sum_1^k A_j$
  – Remaining work after time k: $D(k)$
- Direct $D(k)$ analysis is difficult: Instead imagine a system with always active server
- This imaginary work at time k: $W(k) = N(k)/\mu - k$ (can be negative)

Brownian motion



Definition of Brownian motion

$$B^r(t) = \frac{\sum_1^{r^2 t} \Delta_i}{r} \qquad B(t) = \lim_{r \to \infty} B^r(t)$$

- By central limit theorem, $B(t) \sim \mathcal{N}(0, \sigma^2 t)$
- Independent increments over disjoint intervals
- $B(t_2) - B(t_1) \sim \mathcal{N}(0, \sigma^2 |t_2 - t_1|)$
- $B(t)$ is called a standard Brownian motion $\mathcal{B}_{0,\sigma^2}(t)$
- $B(t) + \theta t + c$ is Brownian motion with drift θ and shift c

Heavy traffic

- Assume work arrivals rate equals the server capacity: $\lambda/\mu = 1$
- **Heavy traffic scaling**: shrink time by r^2 and space by r

$$w^r(t) = W(r^2 t)/r = \frac{N(r^2 t)/\mu - r^2 t}{r}$$

- Difficult to directly analyze imaginary work after time k
- Analyzing imaginary work after k arrivals is easier:
  $V(k) = W(T(k)) = k/\mu - \sum_{i=1}^{k} A_i$
- Heavy traffic scaling of $V(k)$

$$v^r(t) = W(T(r^2 t))/r = \frac{\sum_1^{r^2 t}(1/\mu - A_i)}{r}$$

Note that $1/\mu - A_i$ are i.i.d. and $E\,[1/\mu - A_i] = 1/\mu - 1/\lambda = 0$).

Hence $v^r(t)$ tends to a Brownian motion $\mathcal{B}_{0,a^2}(t)$

Coming back to $w(t)$ from $v(t)$

Intuition: Distribution after large number of arrivals should be similar to that after large time

- Limit theorem for renewal processes:

$$\frac{N(r^2t)}{r^2t} \stackrel{a.s.}{\longrightarrow} \lambda \quad \text{and} \quad \frac{T(r^2t)}{r^2t} \stackrel{a.s.}{\longrightarrow} 1/\lambda$$

- Rewrite $w^r(t)$ to use this fact

$$w^r(t) = \frac{1}{r}\left(\frac{N(r^2t)}{r^2t}\frac{r^2t}{\mu} - T(r^2t)\frac{r^2t}{T(r^2t)}\right)$$

  Hence $w^r(t)$ also tends to a Brownian motion

- Actual work $D(k)$ is given by: $D(k) = W(k) - \min_{0\le i\le k} W(i)$

$$\Rightarrow d^r(t) = w^r(t) - \min_{\tau\in[0,t]} w^r(\tau)$$

  Thus actual remaining work $d(t)$ is a reflected Brownian motion

Remark If $\lambda/\mu = 1-\delta$, Brownian motion of $w(t)$ has drift $-\infty$ and $d(t) = 0$ at all times.

State-space collapse

A two-queue single-server system

- Unit size packets arrive at queue i at rate λ_i − two independent renewal processes
- Server serves one queue at a time- one packet takes unit time
- Heavy traffic: work arrival rate $(\lambda_1 + \lambda_2) \cdot 1 = 1$.
- Queue-lengths at time k are $Q(k) = [Q_1(k), Q_2(k)]$.
- Queue-lengths in heavy traffic scaling: $q^r(t) = Q(r^2 t)/r$

State-space collapse

Let $q^r(t_0) = [a, b]$ and let $[c, d]$ be such that $c + d = a + b$.

- Queue-state can be shifted to $[c, d]$ instantaneously. Proving for $[c, d] = [0, a + b]$ is enough.
- Server starts serving first queue till its empty.

$q_1^r(t)$	$Q_1(r^2 t)$	Actual time needed	Heavy traffic time needed
a	ra	0	0
$\lambda_1 a$	$r\lambda_1 a$	ra	a/r
$\lambda_1^2 a$	$r\lambda_1^2 a$	$r\lambda_1 a$	$\lambda_1 a/r$
$\lambda_1^3 a$	$r\lambda_1^3 a$	$r\lambda_1^2 a$	$\lambda_1^2 a/r$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
0	≈ 0	$\approx \frac{ra}{1-\lambda_1}$	$\frac{1}{r}\frac{a}{1-\lambda_1}$

During this time, $\frac{ra\lambda_2}{1-\lambda_1} = ra$ new packets arrived in second queue.

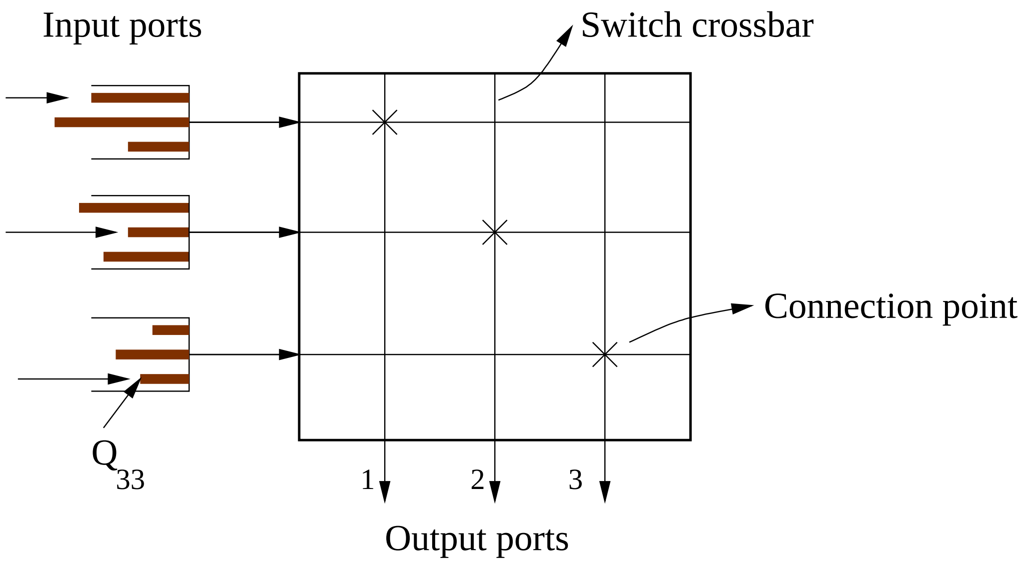
Thus new queue-length q_2^r equals $b + a$.

Time required (in heavy traffic scaling), $\frac{1}{r}\frac{a}{1-\lambda_1}$ vanishes as $r \to \infty$.

State-space Collapse

- Any two states of the same sum queue-length are equivalent, as they can be switched instantly.
- Hence the system is completely described by $q_1(t) + q_2(t)$.
- Equivalent to a single queue system in heavy traffic $\Rightarrow$ $q_1(t) + q_2(t)$ is a reflected Brownian motion
- More generally, let the stability constraint is: $\sum \xi_i \lambda_i \leq c$
- In heavy traffic, i.e. $\sum \xi_i \lambda_i = c$, instant switch between q and $\hat{q}$ if $\sum \xi_i q_i = \sum \xi_i \hat{q}_i$.
- Matrix case $\xi\lambda \leq c$: state-space collapse to more than one dimensions

Switch properties



- Unit size packets arrive at each queue (i, j) at rate λ_{ij}
- Connects each input port to only one output port and vice versa: any permutation (or matching) π denotes one such choice
- At most one packet can be served at each (input/output) port in unit time.

$$\sum_k \lambda_{ik} \leq 1 \quad \text{and} \quad \sum_k \lambda_{kj} \leq 1 \quad \forall i, j$$

Any matrix λ satisfying these constraints is a *stable rate matrix*

Switch dynamics

- Queue-state at time k is $Q(k)$
- Arrivals *at* time k are $A(k)$ (a matrix)
- Departures in interval $[k, k+1)$ be $D(k)$

$$Q(k+1) = Q(k) - D(k) + A(k+1)$$

- The matching chosen between $[k, k+1)$ be $\pi(k)$

A packet departs only if it existed: $D_{ij}(k) = \pi_{ij}(k) 1_{\{Q_{ij}(k)>0\}}$

- Total arrivals up to time k: $\bar{A}(k) = \sum_{i=1}^{k} A(i)$
- Total departures up to time k: $\bar{D}(k) = \sum_{i=1}^{k-1} D(i)$
- $\bar{P}_\pi(k)$: Number of times matching π was used up to time k.

$$D_{ij}(k) = \sum_{\pi} \pi_{ij} 1_{\{Q_{ij}(k)>0\}} (\bar{P}_\pi(k+1) - \bar{P}_\pi(k))$$

Switch Dynamics in Fluid Scaling

Complete description of switch operation: $X(k) = (Q(k), \bar{A}(k), \bar{D}(k), \bar{P}(k))$

- **Fluid scaling** Shrink space and time both by r: $X^r(t) = X(r^2 t)/r$.
  Limit of $X^r(t)$ is the fluid limit $\hat{x}(t) = (\hat{q}(t), \hat{a}(t), \hat{d}(t), \hat{p}(t))$.
- Convert discrete-time dynamics to fluid dynamics. Almost surely,

$$\begin{aligned}
\hat{a}_{ij}(t) &= \lambda_{ij} t \quad \text{i.e.} \quad \hat{a}(t) = \lambda t \\
\hat{q}(t) &= \lambda t - \hat{d}(t) \\
\dot{\hat{d}}_{ij}(t) &= \sum_{\pi} \pi_{ij} 1_{\{\hat{q}_{ij}(t)>0\}} \, \dot{p}_{\hat{\pi}}(t) \\
\text{Define service rate } \sigma(t) &= \sum_{\pi} \pi \dot{p}_{\hat{\pi}}(t)
\end{aligned}$$

$$\begin{aligned}
\dot{\hat{q}}_{ij}(t) &= \lambda_{ij} - \sigma_{ij}(t) && \text{if} \quad \hat{q}_{ij} > 0 \\
&= (\lambda_{ij} - \sigma_{ij}(t))^+ && \text{if} \quad \hat{q}_{ij} = 0
\end{aligned}$$

(A water container with input tap of rate λ_{ij} and output tap rate $\sigma(t)$).

Matrix shorthand for above function is: $\dot{\hat{q}}(t) = (\lambda - \sigma(t))^{+[\hat{q}=0]}$.

Maximum Weight Matching Algorithms

A maximum weight matching algorithm (called MWM-f) chooses a matching π^*, which maximizes the weight

$$\sum_{ij} \pi_{ij} f(Q_{ij}) = f(Q) \cdot \pi \overset{\Delta}{=} \alpha_f(\pi, Q) \text{ over all } \pi$$

Assume the weight function f is a strictly increasing continuous function and $f(0)$ equals zero.

We want optimal matchings for $Q(.)$ be also optimal for $Q(.)/r$. Hence for $(x_1, \cdots, x_n)$ and $(x_1, \cdots, x_n)$ in $\mathcal{R}_+^n$,

$$\sum_i f(x_i) \geq \sum_i f(y_i) \Leftrightarrow \sum_i f(\delta x_i) \geq \sum_i f(\delta y_i) \quad \forall \delta > 0$$

We will show that all such algorithms are stable. Thus even without knowing the arrival rate λ, switch becomes stable.

By stability, we mean $\hat{q}(0) = 0$ implies $\hat{q}(t) = 0$ for all t when λ is stable rate matrix. (Empty containers remain empty).

Thus $\hat{d}(t) = \hat{a}(t)$ at all times.

Stability analysis

Some properties of the service rate $\sigma(t)$ for MWM-f

- At any time t and queue-state $\hat{q}(t)$, suboptimal matchings are not being used, so $\dot{\hat{p}}_\pi(t) = 0$ for them.
- Hence the service rate

$$\sigma(t) = \sum_{\pi} \pi \dot{\hat{p}}_\pi(t) = \sum_{\pi \in \pi^*(t)} \pi \dot{\hat{p}}_\pi(t)$$

  where $\pi^*(t)$ denotes the set of optimal matchings at time t.

- Hence we have (Proof on white-board):

$$f(\hat{q}(t)) \cdot \sigma(t) = \alpha_f^*(\hat{q}(t))$$

  Note that for all matchings π: $f(\hat{q}(t)) \cdot \pi \leq \alpha_f^*(\hat{q}(t))$

Using Lyapunav theory

Consider this (Lyapunav) function of the queue-state $\hat{q}$

$$L(\hat{q}) = \sum_{i,j} F(\hat{q}_{ij}) \ (= F(\hat{q}) \cdot 1) \quad \text{where} \quad F(x) = \int_0^x f(y)\ dy$$

- $L(\hat{q}(t))$ cannot increase over time. (Proof on board)
- Note that $L(\hat{q}(0)) = 0$ if $\hat{q} = 0$. Now $L(\hat{q}(t))$ can not increase nor decrease below 0.
- $L(\hat{q}(t))$ remains zero forever, so does $\hat{q}(t)$. Stability proved.

MWM-f: Steady States in Fluid Scaling

A queue-state q is a steady state (or an invariant state) if $\hat{q}(t_0) = q$ implies all future $\hat{q}(t) = q$. (e.g. $q = 0$ as proved earlier)

- Only steady state is 0 state if no port in heavy traffic. (Everything is drained out finally.)
- If an input/output port is in heavy traffic, its sum queue-length cannot decrease.
  (Again, imagine a water container with input tap rate 1).

  Let input port 1 be in heavy traffic for example, i.e. $\sum_k \lambda_{1k} = \lambda_{1\cdot} = 1$.

$$\dot{\hat{q}}_{1\cdot}(t) \geq \lambda_{1\cdot} - \sigma_{1\cdot}(t) = 1 - \sigma_{1\cdot}(t) = 0$$

- **Two constraints on any trajectory** $\hat{q}(t)$
  - $L(\hat{q}(t))$ cannot increase over time.
  - For ports in heavy traffic, $\hat{q}_{i\cdot}(t)$ or $\hat{q}_{i\cdot}(t)$ cannot decrease.

MWM-f: Steady States in Fluid Scaling

- $L(\hat{q})$ is a strictly convex function of queue-state
- Any future state $\hat{q}(t)$ for initial state q lies in a convex region

$$\hat{q}_{i\cdot}(t) \geq q_{i\cdot} \text{and} \quad \hat{q}_{\cdot j}(t) \geq q_{\cdot j} \quad \text{at heavy traffic ports}$$

- $L(\hat{q})$ has a unique minima for a given initial state
- Since $L(\hat{q}(t))$ keeps decreasing, it lands at the minima eventually.
- If initial state itself is that minima, its a steady state.

Theorem 1 *q is a steady state if and only if q itself is the solution to the optimization problem based on q*

$$\min L(r) \quad \textit{s.t. } r_{i\cdot} \geq q_{i\cdot}, \; r_{\cdot j} \geq q_{\cdot j} \quad \textit{at heavy traffic ports}$$

- **Time of convergence to a steady state**: For arbitrarily small $\epsilon > 0$ and any initial state $\hat{q}(0)$, the queue-state $\hat{q}(t)$ goes within an ϵ-neighborhood of a steady state q within some finite time $T(\epsilon)$.

Heavy traffic scaling

- Recall heavy traffic scaling: $x^r(t) = X(r^2 t)/r$
- The fluid scaling was $X^r(t) = X(rt)/r$, hence $x^r(t) = X^r(rt)$.

  Each instant in heavy traffic scaling is a long period in fluid scaling.
- Fluid process "shortly" converges to a steady state.

  Hence every instant of heavy traffic scaling is in some steady state.

(Different instants can be in different steady states.)

More precisely...

- For studying heavy-traffic scaling over interval $[0, T]$, divide it into r intervals.
- Expand each interval r times and get a fluid scaling process in $[0, T]$
- This fluid process is essentially always in steady state if $T \gg T(\epsilon)$.
- Hence the heavy traffic scaling is also in steady state (esentially always).
- **Caution:** In heavy traffic, steady state does not mean the same as in fluid scaling.

  – Queue-states are a reflected Brownian motion in heavy traffic scaling

  – Now a steady state simply means a solution to the optimization problem in Theorem 1

State space collapse again

- This optimization problem is described by $q_{i\cdot}(t)$ and $q_{\cdot j}(t)$ at heavy traffic ports.
- Corresponding steady state is the unique solution of this problem.
- Thus $q_{i\cdot}(t)$ and $q_{\cdot j}(t)$ at heavy traffic ports completely describe $q(t)$.
- Hence state-space dimension collapses to the number of ports in heavy traffic from n^2.

Single port in heavy traffic

- Say input port 1 is in heavy traffic.
- Given $q_{1\cdot}(t) = a$, determine the entire state $q(t)$.

$$\min \sum_{i,j} F(\hat{q}_{ij}) \text{ such that } \hat{q}_{1\cdot} \geq a$$

- Make all rows zero other than the first.
- Since $L(\hat{q})$ is a symmetric convex function, choose all first row entries equal i.e. a/n. (Jensen's inequality)
- More generally, if neither input port i nor output port j are in heavy traffic, $q_{ij}(t) = 0$ at all times.
- Recall that $q_{1\cdot}(t)$ performs a reflected Brownian motion.

Cost minimization

- Let each unit time cost $\sum_{i,j} F(Q_{ij}(k))$
- We saw MWM-f minimizes this cost at all times in heavy traffic (hence coarser) scaling.
- In practice, minimizing delay is often of interest: minimize $\sum_{i,j} Q_{ij}(k)$
- $f(x) = 1_{\{x>0\}} \stackrel{\Delta}{=} x^0$ should be used.

  This f is not strictly increasing, as needed for stability.

MWM-β Algorithms

- Choose the π maximizing $\pi \cdot Q^\beta$ for some $\beta \geq 0$.
- MWM-0 is same as maximum size matching (unstable).
- MWM-1 is the traditional maximum weight matching–queue-lengths directly used as weights.

All ports in heavy traffic

MWM-$0+$ algorithm

- Slight modification of MWM-0
- For small values of β: $Q_{ij}^{\beta} \approx 1 + \beta \log Q_{ij}$
- Empty queues weigh 0 and non-empty are almost 1.
- Amongst all maximum size matchings, choose the one with maximum $\sum \log Q_{e_i}$ i.e. matching with maximum product of queue-lengths.

MWM-$0+$ is optimal in heavy traffic scaling.

Delay analysis using state-space collapse space

Let $\tilde{q}(t)$ denote the state-vector: vector of all $q_{i\cdot}(t)$ and $q_{\cdot j}(t)$.

- For MWM-$0+$, the state $\tilde{q}(t)$ lies in entire $\mathcal{R}_+^{2n}$.
- Hence it never idles, so delay optimal.
- For MWM-1, the state vector $\tilde{q}(t)$ lies in a proper subspace $\mathcal{S}_1$ of $\mathcal{R}_+^{2n}$.
- Hence idling happens and delay is larger than MWM-$0+$ (contrary to a queueing folklore)
- For $\beta_2 > \beta_1$, state-space of MWM-β_2 is contained in state-space of MWM-β_2. Hence MWM-β_2 has larger delay.