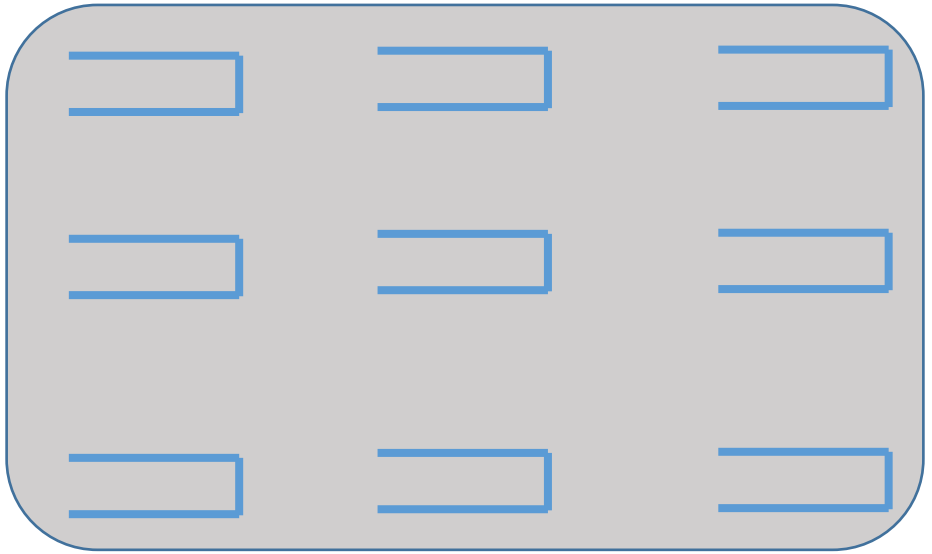
Performance Analysis of Scheduling Algorithms for Switches and Data Centers: Part II

Atilla Eryilmaz, Siva Theja Maguluri, and R. Srikant

OSU, IBM TJ Watson, and UIUC

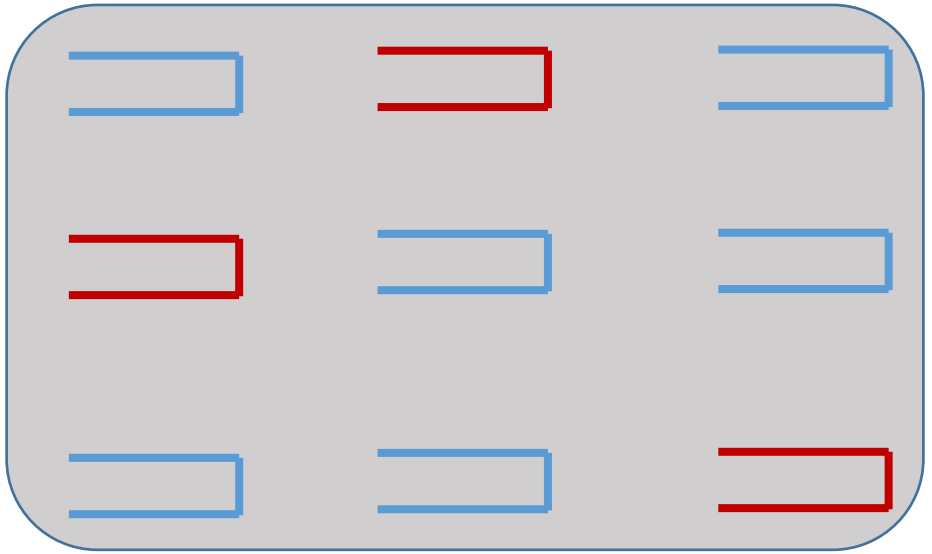
Recap: nxn Switch

- A matrix of queues operating in discrete-time; packets arrive to each queue according to some arrival process. In each time slot, at most one packet can be served from each queue
- Key constraint: At most one queue from each row, and one from each column can be served in each time slot



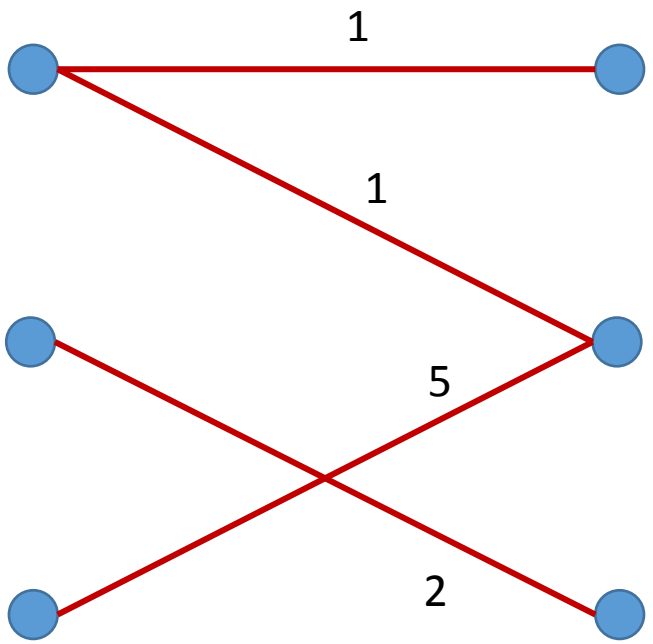
Recap: nxn Switch

- A matrix of queues operating in discrete-time; packets arrive to each queue according to some arrival process. In each time slot, at most one packet can be served from each queue
- Key constraint: At most one queue from each row, and one from each column can be served in each time slot



Throughput-Maximizing Scheduling Algorithm

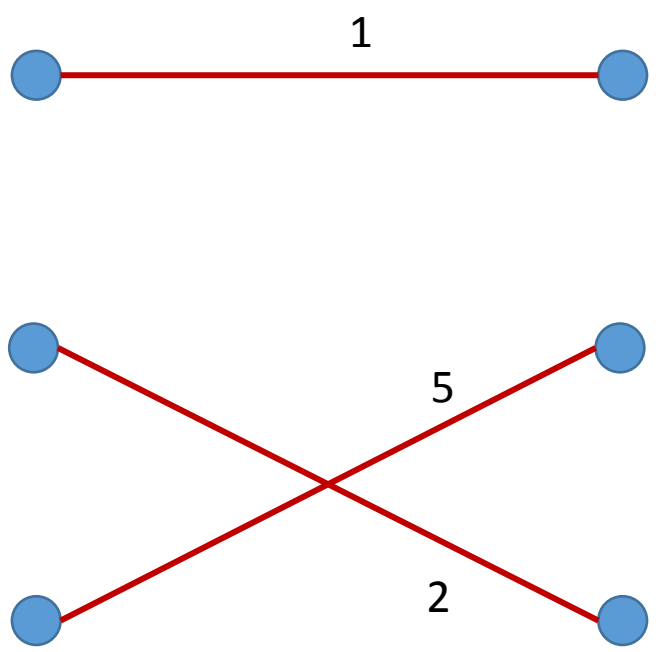
- Bipartite graph: weight of an edge from node i to node j on the right equal to q_{ij}



Throughput-Maximizing Scheduling Algorithm

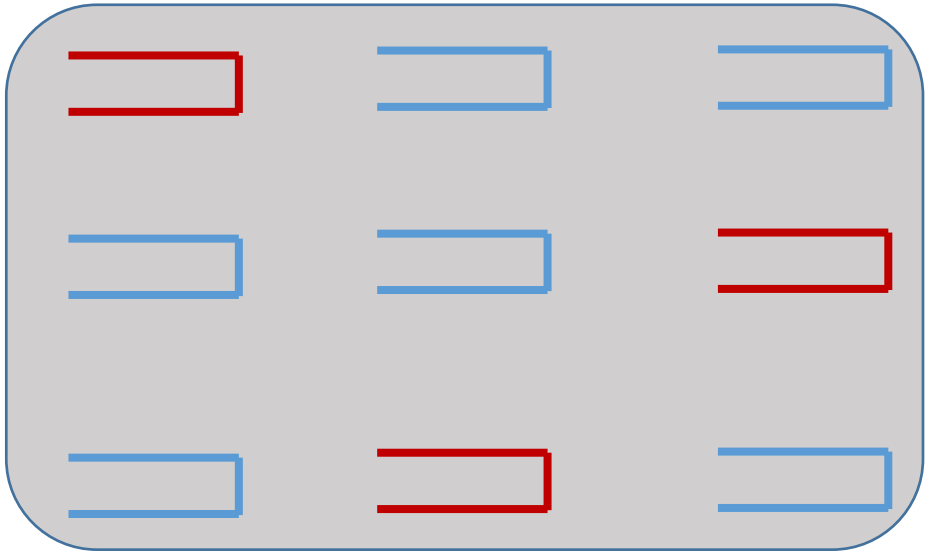
- Bipartite graph: weight of an edge from node i to node j on the right equal to q_{ij}

- Find a matching with the largest weight



Throughput-Maximizing Scheduling Algorithm

- Bipartite graph: weight of an edge from node i to node j on the right equal to q_{ij}
- Find a matching with the largest weight
- Schedule the corresponding queues



Scaling Questions

- Let the total arrival rate to each row and each column be $1 - \epsilon$:

$$\sum_i \lambda_{ij} = 1 - \epsilon, \qquad \sum_j \lambda_{ij} = 1 - \epsilon$$

- Increase the size of the switch n and/or decrease ϵ, how does the expected steady-state queue length scale?

Conjecture 1

$$\frac{K_1 n}{\epsilon} \leq E\left(\sum_{ij} q_{ij}\right) \leq \frac{K_2 n}{\epsilon}$$

- For a fixed ϵ, the delay is $O(1)$, independent of the size of the network

- Unsolved

Conjecture 2: Heavy Traffic

$$\lim_{\epsilon \to 0} \epsilon E\left(\sum_{ij} q_{ij}\right) = \Theta\,(n)$$

- In other words,

$$E\left(\sum_{ij} q_{ij}\right) = \frac{\Theta(n)}{\epsilon} + o\left(\frac{1}{\epsilon}\right)$$

- It is also useful to know the exact expression for the leading term, and if it is optimal.

Conjecture 3

- Suppose $\epsilon = 1/n^{\beta}$
- Then,

$$E\left(\sum_{ij} q_{ij}\right) = \Theta\left(\frac{n}{\epsilon}\right) = \Theta(n^{\beta+1})$$

- We have answers to Conjectures 2 and 3.

Uniform Traffic and Heavy-Traffic

- Assumptions:
  - Uniform Traffic: $\lambda_{ij} = \frac{1-\epsilon}{n}, \forall i, j$
  - Heavy-Traffic Regime: $\epsilon \to 0$.

Theorem: In steady-state, $\lim_{\epsilon \to 0} \epsilon \, E\left(\sum_{ij} q_{ij}\right) = n - \frac{3}{2} + \frac{1}{2n}$

Corollary: $\lim_{\epsilon \to 0} \epsilon \, E(Delay) = 1 - \frac{3}{2n} + \frac{1}{2n^2}$

Non-Uniform Traffic

- Result can be extended to the case of non-Bernoulli arrivals (but still independent across timeslots) and nonuniform traffic

$$\lim_{\epsilon \to 0} \epsilon \, E\left(\sum_{ij} q_{ij}\right) = \left(1 - \frac{1}{2n}\right) ||\sigma||^2$$

  - σ_{ij}^2 is the variance in the number of arrivals/time slot to Queue (i,j)
- Every row and every column is still assumed to be saturated in heavy traffic

Related Work

- Heavy-traffic optimality under a condition called Complete Resource Pooling (CRP)
  - Stolyar (2004); Eryilmaz and S. (2012)
  - One-dimensional state-space collapse
  - This work builds on the second paper above
- State-space collapse and Diffusion Limit without CRP
  - Andrews, Jung, and Stolyar (2007); Shah and Wischik (2012); Wu (2012); Kang and Williams (2012)
  - Multi-dimensional state-space collapse
- Other policies which achieve optimal or near-optimal scaling
  - Neely, Modiano and Cheng (2007); Shah, Walton and Zhong (2012); Shah, Tsitsiklis and Zhong (2014)

Outline of the Proof

- It’s all about unused service
  - Digression 1: Kingman bound for discrete-time queues
  - Digression 2: Heavy-Traffic Optimality of the Join-the-Shortest-Queue (JSQ) Policy
  - Back to Heavy-Traffic Behavior of the MaxWeight Algorithm in a Switch

Kingman Bound for Discrete-Time Queues

Kingman Bound

- $q(k+1) = q(k) + a(k) - s(k) + u(k)$
- Note: $q(k+1)u(k) = 0 \quad \forall k$

In each time slot k,
a(k): # arrivals
s(k): # potential departures
u(k): unused service

- In steady-state, $E(q^2(k)) = E(q^2(k+1))$
- Yields

$$E(q) = \frac{E(a-s)^2}{2(\mu-\lambda)} - \frac{E(u^2)}{2(\mu-\lambda)}$$

Will show that this term is small compared to the first term when $\lambda \to \mu$

Kingman Bound

$$0 \leq \frac{E(u^2)}{2(\mu-\lambda)} \leq \frac{E(u)S_{max}}{2(\mu-\lambda)}$$

S_{max} is the max number of packets that can be served in a time slot

$$E(q) = E(q + a - s + u) \quad \Rightarrow \quad E(u) = \mu - \lambda$$

$$0 \leq \frac{E(u^2)}{2(\mu-\lambda)} \leq \frac{S_{max}}{2}$$

Facts About Unused Service

$$q(k+1)u(k) = 0$$

Main message of this talk: It is useful to view heavy-traffic theory as a generalization of this statement

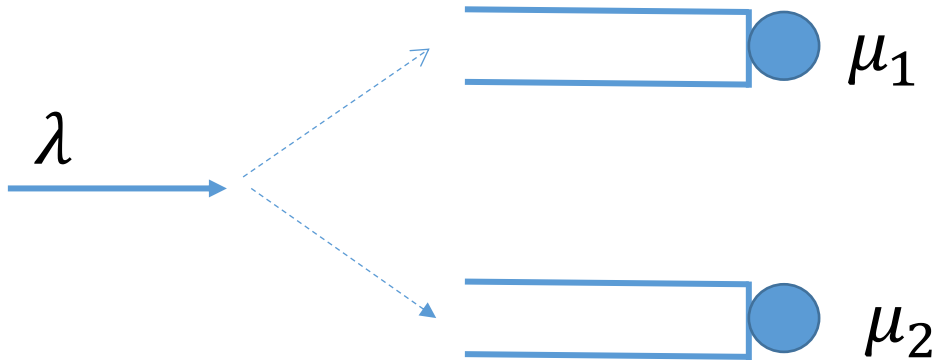
$$0 \leq \frac{E(u^2)}{2(\mu - \lambda)} \leq \frac{S_{max}}{2}$$

Join-the-Shortest-Queue Routing Policy

$$(q_1(k+1) + q_2(k+1))u_1(k) \approx 0$$

JSQ

- Discrete-time model
- Route packet arrivals in each time slot to the shorter of the two queues, breaking ties at random
- Well known that JSQ is heavy-traffic optimal (Foschini and Gans, 1978); will derive this result using the Kingman-type drift argument



Lower Bound – Resource Pooling

- Queue length is smallest if both servers act as one

- Kingman bound:

$$E(q_1 + q_2) \geq \frac{E(a - s_1 - s_2)^2}{2(\mu_1 + \mu_2 - \lambda)} - \frac{S_{max1} + S_{max2}}{2}$$

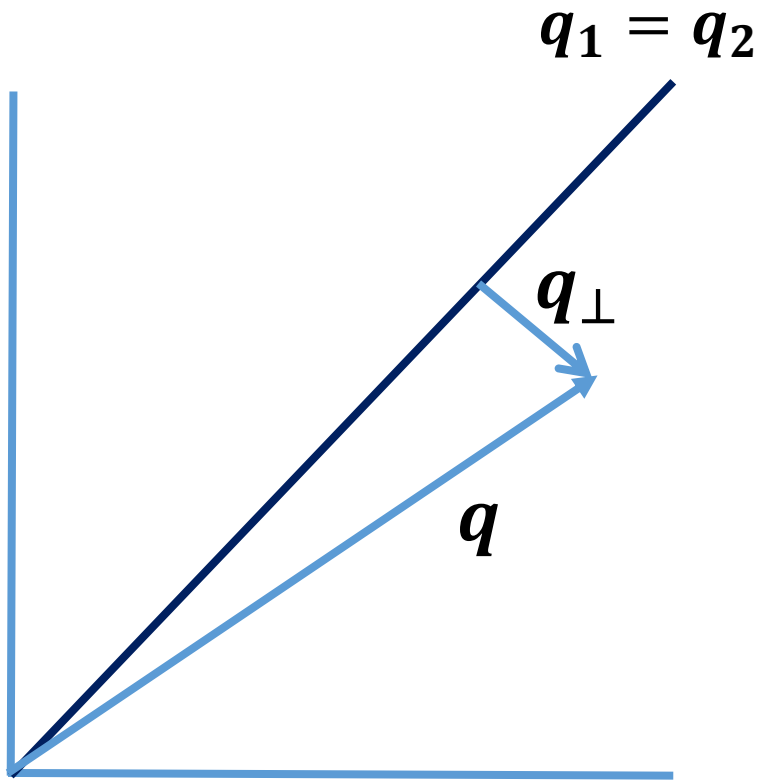
State-Space Collapse

- For resource pooling to occur under JSQ, we need $q_1 \approx q_2$.
  - There is no unused service (idling) in one queue when there is work in the other queue

- In our definition, this means

$$E(||q_\perp||^2) \leq M,$$

where M does not depend on the heavy-traffic parameter $\epsilon = \mu_1 + \mu_2 - \lambda$.



Upper bound for JSQ

- Set the drift of $V(q) = (\sum_l q_l)^2$ equal to zero:

$$E(V\big(q(k+1)\big) - E(V\big(q(k)\big) = 0$$

- Why this choice of $V(q)$?
  - From the state-space collapse result, we expect the queues to behave like a single queue as in the lower bound; all queues are roughly equal, so they would all hit zero simultaneously
  - So we expect $\sum_l q_l$ to behave like a single-server queue

Using State-Space Collapse

- The terms in the drift equation look very similar to the lower bound, except terms of the form:

$$(q_1(k+1) + q_2(k+1))u_1(k).$$

- Note that $q_1(k+1)u_1(k) = 0$, but $q_2(k+1)u_1(k) \neq 0$

- But, from state-space collapse, $q_1(k+1) \approx q_2(k+1)$, and thus, $q_2(k+1)u_1(k) \approx 0$

Back to MaxWeight Scheduling in a Switch

$$u_{ij}(k)\left(\sum_{j'} q_{ij'}(k+1) + \sum_{i'} q_{i'j}(k+1) - \frac{1}{n}\sum_{i'j'} q_{i'j'}(k+1)\right) \approx 0$$

Back to the Switch Problem

- Recall that we assume $\sum_i \lambda_{ij} = 1 - \epsilon,\ \sum_j \lambda_{ij} = 1 - \epsilon$

- As $\epsilon \to 0$, this is equivalent to $< \lambda, e_i > = 1, < \lambda, \tilde{e}_j > = 1$
  - e_i is a matrix with 1's in the i^{th} row and zeros everywhere else
  - $\tilde{e}_j$ is a matrix with 1's in the j^{th} row and zeros everywhere else

- The arrival rate matrix λ lies on the intersection of the hyperplanes with normal vectors $e_1, \dots, e_n, \tilde{e}_1, \dots, \tilde{e}_n$

$$e_i = i^{th}\ row \begin{bmatrix} 0 & 0 & 0 \\ & \vdots & \\ 1 & 1 & 1 \\ & \vdots & \\ 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{e}_j = \begin{bmatrix} 0 & & 1 & & 0 \\ 0 & \dots & 1 & \dots & 0 \\ 0 & & 1 & & 0 \end{bmatrix}$$

$j^{th}\ col$

Back to the Switch Problem

- Recall that we assume $\sum_i \lambda_{ij} = 1 - \epsilon,\ \sum_j \lambda_{ij} = 1 - \epsilon$

- As $\epsilon \to 0$, this is equivalent to $< \lambda, e_i > = 1, < \lambda, \tilde{e}_j > = 1$

$$e_i = i^{th}\ row \begin{bmatrix} 0 & 0 & 0 \\ & \vdots & \\ 1 & 1 & 1 \\ & \vdots & \\ 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{e}_j = \begin{bmatrix} 0 & & 1 & & 0 \\ 0 & \dots & 1 & \dots & 0 \\ 0 & & 1 & & 0 \end{bmatrix}$$

$j^{th} col$

State Space Collapse for Switch

- The queue length vector collapses to the cone spanned by the normal vectors $e_1, \ldots, e_n, \tilde{e}_1, \ldots, \tilde{e}_n$
- Any vector in the cone is of the form $w_1 e_1 + \cdots + w_n e_n + \tilde{w}_1 \tilde{e}_1 + \cdots + \tilde{w}_n \tilde{e}_n$

$$q = \begin{array}{c} \\ w_1 \\ \vdots \\ w_i \\ \vdots \\ w_n \end{array} \begin{array}{c} \begin{array}{cccccc} \tilde{w}_1 & & \cdots & \tilde{w}_j & \cdots & \tilde{w}_n \end{array} \\ \begin{bmatrix} w_1 + \tilde{w}_1 & & w_1 + \tilde{w}_j & & w_1 + \tilde{w}_n \\ & & \vdots & & \\ w_i + \tilde{w}_1 & \cdots & w_i + \tilde{w}_j & \cdots & w_i + \tilde{w}_n \\ & & \vdots & & \\ w_n + \tilde{w}_1 & & w_n + \tilde{w}_j & & w_n + \tilde{w}_n \end{bmatrix} \end{array}$$

Upper Bound for Switch

- No resource pooling
- Set the drift of

$$V(q) = \sum_i \left(\sum_j q_{ij} \right)^2 + \sum_j \left(\sum_i q_{ij} \right)^2 - \frac{1}{n} \left(\sum_{ij} q_{ij} \right)^2$$

  equal to zero

- The first two terms would correspond to “resource pooling” along each row and column
  - The first term in parenthesis is the sum of the queue lengths in row i
  - The second term in parenthesis is the sum of the queue lengths in column j
  - The third term in parenthesis is the total queue length in the switch

Why V(q)?

- Unused service term in the drift expression provides the clue

$$u_{ij}(k)\left(\sum_{j'} q_{ij'}(k+1) + \sum_{i'} q_{i'j}(k+1) \;-\; \frac{1}{n}\sum_{i'j'} q_{i'j'}(k+1)\right)$$

$$= n u_{ij}(k)\left(w_i + \tilde{w}_j\right) = n u_{ij}(k) q_{ij}(k+1)$$

Using $q_{ij} = w_i + \tilde{w}_j$

= 0 (This expression will not be zero without the third term above!!!)

Back to Main Result

- Use $E(V\big(q(k+1)\big) = E(V(q(k))$

- And the result from the previous page to obtain

Theorem: In steady-state, $\lim_{\epsilon \to 0} \epsilon\, E(\sum_{ij} q_{ij}) = n - \frac{3}{2} + \frac{1}{2n}$

Corollary: $\lim_{\epsilon \to 0} \epsilon\, E(Delay) = 1 - \frac{3}{2n} + \frac{1}{2n^2}$

Joint Scaling of Traffic and Switch Size

- Scaling the traffic parameter $\epsilon \to 0$ with $n \to \infty$ as $\epsilon = n^{-\beta}$
- Conjecture becomes: $E\left(\sum_{ij} q_{ij}\right) = O\left(\frac{n}{\epsilon}\right) = O\left(n^{1+\beta}\right) \quad \forall \beta > 0$

Theorem: In steady-state, for $\beta > 4$, $\quad E\left(\sum_{ij} q_{ij}\right) = O(n^{1+\beta})$

Conclusions

- An exact heavy-traffic formula for the stationary total queue length in a switch operating under the MaxWeight algorithm
  - Assumption: saturated rows and columns

- It's all about generalizing this equation: $q(k+1)u(k) = 0$

- Higher Moments??? Eryilmaz and S. (2012), Ying and S. (2014)