Optimal Routing Between Alternate Paths With Different Network Transit Delays

Essia H. Elhafsi
Dept. of Computer Science and Engineering
University of California
Riverside, CA. 92521
essia@cs.ucr.edu

Mart Molle
Dept. of Computer Science and Engineering
University of California
Riverside, CA. 92521
mart@cs.ucr.edu

***Abstract*— We consider the path-determination problem in Internet core routers that distribute flows across alternate paths leading to the same destination. We assume that the remainder of the network transit delay beyond this router are different for the two paths, so a good routing policy can reduce the end-to-end delay by favoring the faster path. Thus, we propose and solve the optimal path-determination problem for a router, which minimizes the average network transit delay for a flow by dynamically assigning each packet to one of the available output ports based on their respective instantaneous queue lengths and on the average network transit delay for the associated path. We assume that all outgoing link speeds at the router are equal, but we generalize the model to allow each output port to serve a link group (such as an optical fiber using WDM) that consists of multiple physical channels running in parallel.**

By formulating path-selection as a Markov Decision Problem, we show that the optimal algorithm is a threshold-type policy that we call $JSQ + b$.

keywords: Threshold routing; Markov decision process; Asymmetric paths; Virtual paths; Optical networks.

I. INTRODUCTION

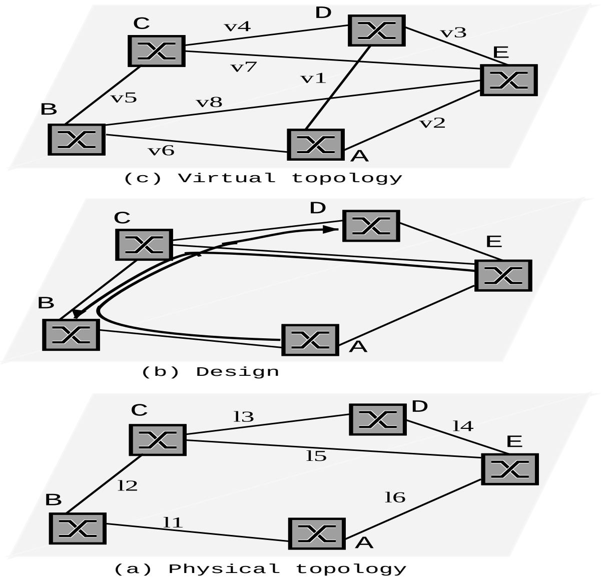
Consider an Internet core router, A, with *two available paths* to the *same destination*, D. Although some routing protocols (e.g., distance-vector, or RIP) restrict A to sending traffic over one of those paths at a time, others (e.g., link-state, or OSPF) support dynamic load-balancing over multiple paths. In that case, we call A a *forking node* with respect to destination D and, naturally, are interested in finding ways by which A can reduce the end-to-end delay for traffic going to D through its local path-selection policy.

This path-selection problem is fundamental to a variety of interesting applications. For example, suppose node A is the boundary router for a "stub" Autonomous System with uplinks to two different Internet Service Providers. Typically, A will route all traffic through one ISP uplink, leaving the second ISP uplink as a backup route to improve reliability. If the traffic bottleneck is far away from A, then single-path routing will have little effect on end-to-end delay as long as A routes all traffic through the faster path. However, since the two uplinks from A are dedicated to a single customer network, they are likely to have very limited bandwidth. Thus, dynamic load-balancing by A across both of its available uplinks may have a significant impact on the end-to-end performance for its network.

Routing in optical networks with WDM links [8], [6] is another application domain where dynamic load balancing may provide a significant performance improvement. In this case, each physical cable attached to a router's output port can carry many independent data streams simultaneously over different wavelengths. More importantly, the physical layer may support optical cross-connection of specific wavelengths between adjacent cables, to create a direct, all-optical, multi-hop *Virtual Link* between two physically non-adjacent routers. For example, consider the routing of traffic from node A to node D in the optical network given in Fig. 1(a). Suppose all AD traffic follows the 3-hop physical path (l_1, l_2, l_3) through intermediate nodes B and C. Nevertheless, we might have the choice between two alternative paths at the logical level, as shown in Fig. 1(c). Path 1 uses Virtual Link v_1 to reach D in a single "router-hop", after we have configured the optical interconnects appropriately at intermediate nodes B and C (Fig. 1(b)). Path 2 requires three "router-hops" to reach D through the logical path $(v_4,\ v_5,\ v_6)$. Since path 1 avoids the store-and-forward delay at intermediate routers B and C, the remainder of network transit delay — between a packet's departure from A and its arrival at D — will be much shorter if we route the packet along path 1 than path 2.

In [5], we introduced the forking node scheduling problem with path-dependent network transit delays. We also compared the performance of a variety of random, deterministic and state-dependent routing algorithms in this application, using simulations. From all the algorithms we tested, a simple generalization of the well-known *Join-the-Shortest-Queue (JSQ)* consistently gave us the best performance according to a variety of metrics. The generalization was simply to add a bias b to favor the queue leading to the faster path. We now provide a further generalization of the $JSQ+b$ policy to handle multi-server link groups (such as WDM optical fibers). In this case, if an arriving packet has a choice between output port 1, which has i waiting packets and is served by a group of m_1 parallel servers, or output port 2, which has j waiting packets and is served by a group of m_2 parallel servers, then the routing policy assigns the arrival to port 1 unless the difference in the normalized queue length per channel, $\frac{j}{m_2} - \frac{i}{m_1} > b$.

We formulate path-determination as a Markov Decision Problem [1], [7], and derive the optimal solution under the



(c) Virtual topology

(b) Design

(a) Physical topology

Fig. 1. The physical, design and virtual topologies. Virtual channel v_1 uses optical links l_1, l_2 and l_3. There is a single hop path $p_1 = (v_1)$ between S and D, as well as a multihop path $p_2 = (v_4,\ v_5,\ v_6)$

Poisson traffic model. In particular, we show that the stationary routing policy at a forking node that minimizes the average end-to-end delay experienced by a packet, is of threshold-type (also known as switch-type), and is in fact $JSQ + b$.

The rest of this paper is organized as follows. In Section II we develop a dynamic programming formulation of the optimal path-selection problem, and present an interative algorithm for finding the optimal solution using Markov decision processes. In Section III, we present some fundamental properties about the optimal policy, from which we show analytically that it is representable as a *switch curve*. In Section IV, we characterize the shape of the switch curve and show that it is identical to our $JSQ + b$ policy. We present numerical results in Section V and conclude in Section VI.

II. Model Description and Formulation

1. Model Description

Node A can be modeled as a queueing system with one input feeding two parallel queues, each connected to its own multi-server output. We assume that the routing decision maker at node A has complete information, i.e., it knows the instantaneous number of packets at each queue, the routing bias b and the mean difference in downstream path delays d. Thus, the structure of the system is that of a Markovian decision process which can be described as follows.

States: The state of the system is described by the tuple (i, j), where i and j are the number of packets in queue 1 and queue 2 respectively.

Events: When an event e occurs, we distinguish three possibilities: $e = 0$ when a packet arrives on the common input; $e = 1$ when a packet departs from queue 1; and $e = 2$ when a packet departs from queue 2.

Decision: Following each event, the decision maker must choose one of the following actions: $a = 0$ means do nothing; $a = 1$ means route a packet to queue 1; or $a = 2$ means route a packet to queue 2. Since every incoming packet must be assigned to one output queue, but we do not allow jockeying between the two queues, the set P of valid state transitions $P^{(\cdot)} = (\Delta i, \Delta j)$ is limited to the following elements: $P^{(+1)} \equiv (+1, 0)$, where $e = 0$, $a = 1$ and the arrival joins queue 1; $P^{(+2)} \equiv (0, +1)$, where $e = 0$, $a = 2$ and the arrival joins queue 2; $P^{(-1)} \equiv (-1, 0)$, where $e = 1$, $a = 0$ and the departure leaves queue 1; and $P^{(-2)} \equiv (0, -1)$, where $e = 2$, $a = 0$ and the departure leaves queue 2.

Criterion: The objective is to minimize T^*, the expected end-to-end packet delay in steady-state.

Costs and rewards: Without loss of generality, we consider only the difference in the mean downstream path delays, d, rather than their actual values, d_1 and d_2. At all times, the decision maker keeps a running total of all costs incurred and rewards received for all packets it has handled so far. For each arriving packet, the decision maker incurs a cost equal to its expected end-to-end delay, conditioned on the state of the system at its arrival. For each departing packet, the decision maker receives a reward equal to the minimum overall average cost, T^*. Therefore, ***if the decision maker follows the optimal policy, then the cumulative expected cost/reward will be zero at the end of each regenerative cycle.***

2. Dynamic Programming Formulation

In the following analysis, we complete the model in terms of a mathematical formulation. We use the following notation:

- $f_n(i, j)$ denotes the cumulative expected cost/reward, given the system has reached state (i, j) following n state changes starting from a randomly-chosen initial state (i.e., $f_0(i, j) \equiv 0$ for all i, j). Although these initial conditions are highly atypical, their effect is not significant when $n \gg 1$.
- $g(i, j : P^{(\cdot)})$ is the incremental cost/reward for responding to event e with action a in state (i, j), leading to state change $P^{(\cdot)} = (\Delta i, \Delta j)$. The incremental cost/reward is:

$$g(i, j : P^{(z)}) = \begin{cases} d + max\left(0, \frac{i+1}{m_1} - 1\right) & if\ z = 1 \\ max\left(0, \frac{j+1}{m_2} - 1\right) & if\ z = 2 \\ -T^* & if\ z = -1, -2 \end{cases}$$

- $\Lambda(i, j : P^{(\cdot)})$ is the departure rate from state (i, j) through state change $P^{(\cdot)} = (e, a)$. Since all interarrival and service times are exponential, we see that:

$$\begin{aligned} \Lambda(i, j : P^{(1)}) &= \lambda \quad \text{if } a = 1, \text{ or } 0 \text{ otherwise} \\ \Lambda(i, j : P^{(2)}) &= \lambda \quad \text{if } a = 2, \text{ or } 0 \text{ otherwise} \\ \Lambda(i, j : P^{(-1)}) &= \min(i, m_1) \\ \Lambda(i, j : P^{(-2)}) &= \min(j, m_2) \end{aligned}$$

The decision maker now has to make a choice between action $a = 1$ leading to state change $P^{(1)}$ and action $a = 2$ leading to a state change $P^{(2)}$ such that the incremental cost is minimized, i. e., $min(f_n(i+1, j) + g(i, j : P^{(1)}), f_n(i, j + 1) + g(i, j : P^{(2)})$. The model can now be defined for all

$i \geq 0$ and all $j \geq 0$ through the following dynamic programming equation (DPE). For simplicity, we define $\Lambda(i,j) = \sum_{P^{(\cdot)} \in P} \Lambda(i, j : P^{(\cdot)})$ as the combined departure rate from state (i,j).

$$\begin{aligned} f_0(i,j) &= 0 \\ f_{n+1}(i,j) &= \sum_{P^{(\cdot)} \in P} \frac{\Lambda(i,j:P^{(\cdot)})}{\Lambda(i,j)} \cdot \\ &\quad \left(f_n(i+\Delta i, j+\Delta j) + g(i,j:P^{(\cdot)})\right) \end{aligned}$$

if $n > 0$, which can be solved iteratively to obtain the optimal policy. Algorithm 1 illustrates the solution procedure where we assume that the cardinality of the state space is $x_1 \times x_2$.

Algorithm 1 Computing the optimal policy

```
$n \Leftarrow 0$
for $i = 1 \to x_1$ do
    for $j = 1 \to x_2$ do
        $f_n(i,j) \Leftarrow 0$
    end for
end for
repeat
    for $i = 1 \to x_1$ do
        for $j = 1 \to x_2$ do
            $h_1 \Leftarrow f_n(i+1,j) + g(i,j:P^{(1)})$
            $h_2 \Leftarrow f_n(i,j+1) + g(i,j:P^{(2)})$
            $\Lambda \Leftarrow \lambda + \min(i,m_1) + \min(j,m_2)$
            if $h_1 < h_2$ then
                $f_{n+1}(i,j) \Leftarrow \lambda \cdot h_1$
                $\Pi(i,j) \Leftarrow 1$
            else
                $f_{n+1}(i,j) \Leftarrow \lambda \cdot h_2$
                $\Pi(i,j) \Leftarrow 2$
            end if
            $f_{n+1}(i,j) \Leftarrow (f_{n+1}(i,j) + \min(i,m_1) \cdot (f_n(i-1,j) - T^*) + \min(j,m_2) \cdot (f_n(i,j-1) - T^*))/\Lambda$
        end for
    end for
    $n \Leftarrow n + 1$
until $\| f_n - f_{n-1} \| < \epsilon$
```

III. Characterization of the Optimal Policy

Theorem 1: If it is optimal to route an arriving packet to queue 1 in state (i,j), then it is optimal to route an arriving packet to queue 1 in all states $(i, j+k)$ and in all states $(i-k, j)$ for $k \geq 0$.

Alternatively, Theorem 1 can be stated as:

Remark 1: if it is optimal to route an arriving packet to queue 2 in state (i,j), then it is optimal to route an arriving packet to queue 2 in all states $(i+k, j)$ and in all states $(i, j-k)$ for $k \geq 0$.

In order to prove Theorem 1, we will prove the monotonicity results given by Proposition 1 where we define

$$\Delta_i f_n(i,j) = f_n(i+1,j) - f_n(i,j)$$
$$\Delta_j f_n(i,j) = f_n(i,j+1) - f_n(i,j)$$

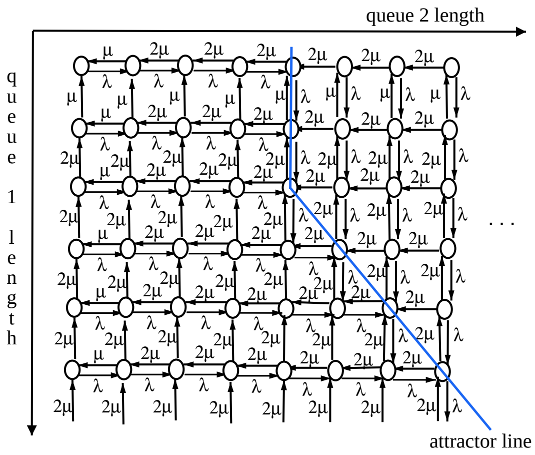


Fig. 2. CTMC with $(m_1, m_2) = (2, 2)$. The *attractor line* is for $b = 2$.

Proposition 1: $\forall n \geq 0$,

$$\Delta_j f_n(i, j+1) \geq \Delta_j f_n(i+1, j) \quad (1)$$
$$\Delta_i f_n(i, j+1) \leq \Delta_i f_n(i+1, j) \quad (2)$$
$$\Delta_j f_n(i+1, j) \geq \Delta_j f_n(i, j) \quad (3)$$

Remark 2: f_n is component-wise convex in i and in j if the following hold:

$$\begin{aligned} \Delta_i f_n(i+1,j) &\geq \Delta_i f_n(i,j) \\ \Delta_j f_n(i,j+1) &\geq \Delta_j f_n(i,j) \end{aligned}$$

Properties (1) and (2) reflect the fact that the difference in delay between having one more packet at queue 1 and having one more packet at queue 2 is non-increasing (nondecreasing) function of the number already in queue 1 (queue 2). Hence we switch from preferring to have an additional packet at queue 1 to having one packet at queue 2 as either the number at queue 1 increases or as the number at queue 2 decreases. Therefore, a switch curve exists. Property (3) is known as the super-modularity property of f_n in (i,j). With some algebraic manipulations, it can be shown that properties (1), (2) and (3) together imply component-wise convexity of f_n in i and in j. We prove the proposition by induction on n and we refer the reader to [3] for a complete proof.

IV. Characterization of the Switch Curve

1. What is an Attractor Line?

Node A can be modeled as a two-dimensional Markov chain whose state space may be represented by the tuple (i,j). The state transition diagram of the system is shown in Fig. 2 where JSQ+b routing policy dictates the transition behavior between the states. The line $\frac{j}{m_2} - \frac{i}{m_1} = b$ in the system state space is called the *attractor line* for the following reasons. Notice that for all states that are *not* on the *attractor line*, an arrival (i.e., an outgoing transition with label λ) always points towards the *attractor line*. Thus, for all states below (to the left of) the *attractor line*, an arrival moves the current state towards the right. Conversely, for all states above (to the right of) the *attractor line*, an arrival moves the current state downwards. The *attractor line* is shown in Fig. 2.

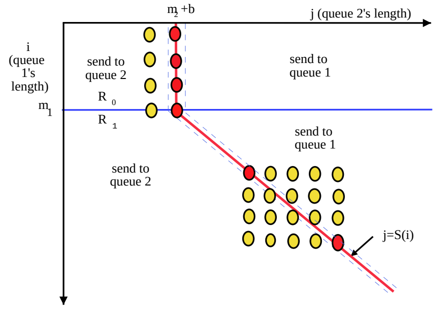


Fig. 3. Switch curve characteristics over the state space. $(m_1, m_2) = (3, 4)$

2. Attractor Line and Switch Curve

Our objective here is to prove that the switch curve, S, which is defined, with respect to j, by Eq. (4) has the same shape as the *attractor line*, namely a straight line (and more precisely a staircase shape).

$$\begin{aligned} S(i) = \quad & min(j \geq 0 :\ f_n(i,j : P^{(1)}) + g(i,j : P^{(1)}) \\ & -f_n(i,j : P^{(2)}) - g(i,j : P^{(2)}) < 0) \end{aligned} \qquad (4)$$

We divide the state space in two regions, a boundary region $R_0 = \{(i,j) : i < m_1,\ j \geq 0\}$ and a homogeneous region $R_1 = \{(i,j) : i \geq m_1, j \geq 0\}$, to show that in region R_0, the switch curve will be a vertical line and in region R_1 it will be a straight line with a nonzero slope as shown in Fig. 3.

3. Shape of the Switch Curve at the Boundary Region

Theorem 2: If the difference in cost between having one more packet in queue 2 and having one more packet in queue 1 at state $(i, m_2 + b - 1)$ is the same as the difference in cost between having one more packet in queue 2 and having one more packet in queue 1 at state $(i + k, m_2 + b - 1)$, for $k \geq 0,\ i + k < m_1,\ j > 0$, then the switch curve at R_0 is a vertical line. $j = m_2 + b - 1$

Theorem 2 can be also stated as:

Remark 3: if it is optimal to route an arriving packet to queue 2 in state $(i, j) = (i, m_2 + b - 1)$, then it is optimal to route an arriving packet to queue 2 in all states $0 \leq i < m_1$ and $0 \leq j \leq m_2 + b - 1$. Conversely, if it is optimal to route a new packet to queue 1 at state $(i, m_2 + b)$ then it is optimal to route a new packet to queue 1 at all states $(i + k, m_2 + b)$ (and in general in all states $(i + k, m_2 + b + k_0),\ k_0 \geq 0$) for all $k \geq 0$ when $0 < i < m_1$.

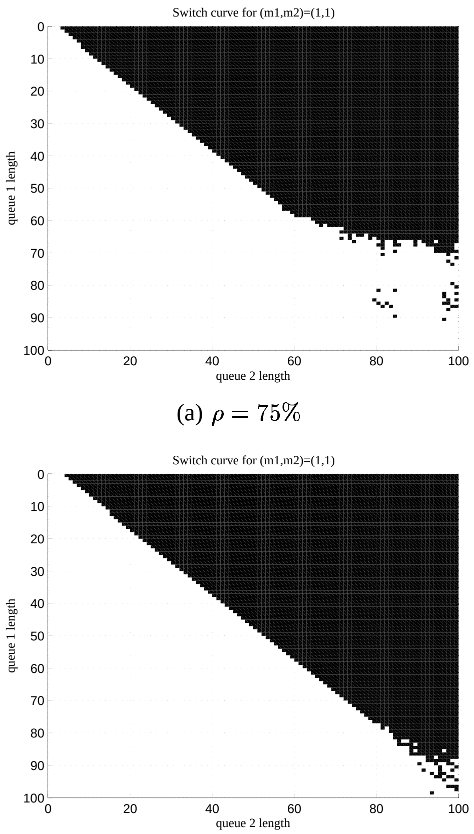
To prove Theorem 2, we prove the following Proposition by induction on n. For the proof we refer the reader to [3].

Proposition 2: $\forall\ n \geq 0,\ 0 \leq i < m_1,\ j \geq 0,$

$$\Delta_i f_n(i, j + 1) = \Delta_i f_n(i + 1, j)$$
$$f_{n+1}(i, j) - 2f_{n+1}(i + 1, j) + f_{n+1}(i + 2, j) = 0$$

4. Shape of the Switch Curve at the Homogeneous Region

Theorem 3: If the difference in cost between having one more packet in queue 1 and having one more packet in queue 2 is the same as the difference in cost between having $m_1 + 1$ more packet in queue 1 and having $m_2 + 1$ more packet in queue 2, then the switch curve can be characterized by its slope which is equal to $\frac{m_2}{m_1}$.



(a) $\rho = 75\%$

(b) $\rho = 60\%$

Fig. 4. Switch curve when $(m_1, m_2) = (1, 1),\ d = 20/\mu$

Remark 4: Alternatively, Theorem 3 can be stated as: if it is optimal to route an arriving packet to queue 1 in state (i, j), then it is optimal as well to route it to queue 1 in all states $(i + m_1, j + m_2)$.

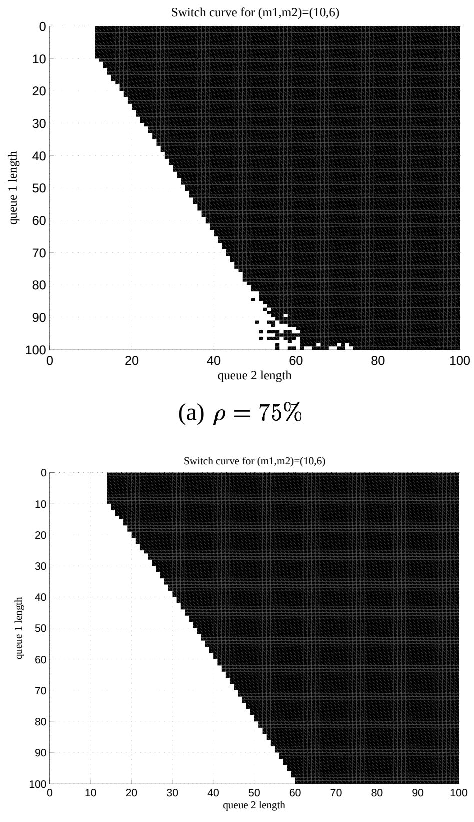
To prove Theorem 3, we prove the following Proposition by induction. We refer the reader to [3] for a complete proof.

Proposition 3: $\forall\ n \geq 0,$

$$\begin{aligned} f_n(i + 1, j) - f_n(i, j + 1) \quad & = f_n(i + m_1 + 1, j + m_2) \\ & \quad -f_n(i + m_1, j + m_2 + 1) \end{aligned}$$

V. Numerical Results

Fig. 4 and Fig. 5 show the switch curves for several link group sizes for a network load of $\rho = 60\%$ and 75%. The switch curves are straight lines that correspond to the models respective *attractor lines*. However, as we move away from the origin especially in the case of $m_1 = m_2 = 1$, the behavior of the switch curves become ambiguous. When $m_1 = 10$ and $m_2 = 6$, the boundary effect is less pronounced. We believe that this is due to the boundary effect and the truncation of the state space. Moreover, we notice that this behavior persists for high loads (see Fig. 4(a) and Fig. 5(a)). For low loads, we observe that the switch curves behave better and they continue to be straight lines throughout the state space (see Fig. 4(b)



(a) $\rho = 75\%$

(b) $\rho = 60\%$

Fig. 5. Switch curve when $(m_1, m_2) = (10, 6),\ d = 200/\mu$

and Fig. 5(b)). To avoid this problem, a larger state space should be considered, keeping in mind that the computation effort increases exponentially as the state space increases.

The switch curves can be used to determine the optimal b for a given arrival rate λ and a downstream delay difference d. We argue that the switch curve intercept (I) is a function of the routing bias i. e., $I = b + m_2$. Moreover, the figures show that $I < 20 \leq d$ $(\mu = 1)$ thus, $b < d$.

Fig. 6 shows that the relative delay difference between having one more packet at queue 1 and having one more packet at queue 2 is a non-increasing function in the number in queue 1. The set of points, s_i, in the curves that cross the x-axis constitute the set of points on the switch curve (s_i is called the switch-over point that is, the routing policy keeps sending packets to queue 2 until its length reaches s_i, at this point it switches and sends packets to queue 1).

VI. Conclusion

We solve the path-selection problem for an Internet core router that chooses from alternate routes leading to the same destination. We prove that the optimal solution is a threshold-type routing policy called $JSQ+b$, in which packets are only routed to the output port leading to the higher-delay path if the difference in output-queue lengths is greater than some threshold b. Moreover, our optimality result for $JSQ+b$ is general enough to handle multiserver link groups for each output port,

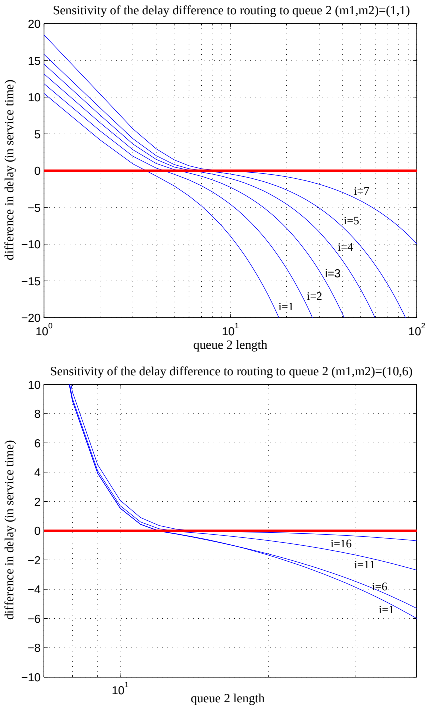


Fig. 6. Delay difference (in log-scale) between having one more packet in queue 1 and having one more packet in queue 2. It is decreasing function of the number in queue 1. Here $(m_1, m_2) \in \{(1,1),\ (10,6)\}$ and $\rho = 75\%$.

in which case each queue length is normalized by the number of available servers before the threshold comparison.

$JSQ + b$ includes the "greedy" policy as a special case, where packets always choose the faster path unless the extra queueing delay incurred at its output port exceeds the savings in network transit delay from taking the faster path. Surprisingly, our work shows that the "greedy" policy is *not* optimal for this problem, which contradicts previous results on this subject [2]. In general, we found that the optimal bias is significantly less than the "greedy" value. In the case of the single server link groups we showed that b is $O(Log(d))$ [4].

References

[1] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena scientific, 1995.

[2] P. Dube, V. S. Borkar, and D. Manjunath. Differential join prices for parallel queues: Social optimality, dynamic pricing algorithms and application to internet pricing. In *Proceedings of IEEE INFOCOM*, 2002.

[3] E. H. Elhafsi. *Modeling forking nodes with state dependent routing: Application to Asymmetric networks*. Ph. D. Dissertation, University of California Riverside, Riverside, CA, USA, 2005.

[4] E. H. Elhafsi, M. Molle, and D. Manjunath. On the application of forking nodes in queueing networks (*submitted to the international journal of communication systems*).

[5] E. H. Elhafsi, M. Molle, and D. Manjunath. Can we use product-form solution techniques in networks with asymmetric paths? In *Proceedings of the International symposium on Performance Evaluation of computer and telecommuniaction systems (SPECTS 2005)*, pages 348–359, 2005.

[6] Ahmed Mokhtar and Murat Azizolu. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Trans. Netw.*, 6(2):197–206, 1998.

[7] J. Walrand. A note on optimal control of a queueing system with two heterogeneous servers. *Systems and Control Letters*, 4:131–134, 1984.

[8] J. M. Yates, M. P. Rumsewicz, and J. P. R. Lacey. Wavelength converters in dynamically-reconfigurable wdm networks. *IEEE Communications Surveys & Tutorials*, 2(2):2–15, Second Quarter 1999.