CS269I: Incentives in Computer Science Lecture #7: Selfish Routing and Network Over-Provisioning*

Tim Roughgarden†

October 17, 2016

This lecture discusses incentive issues in shortest-delay, or "selfish," routing. The mathematical model was originally proposed for road networks (where drivers constitute the traffic), but it is also relevant for communication networks (where data packets constitute the traffic). Shortest-path routing is common in local area networks. Routing in the Internet (between different local networks) is done a bit differently, as we'll see in Lecture #8.

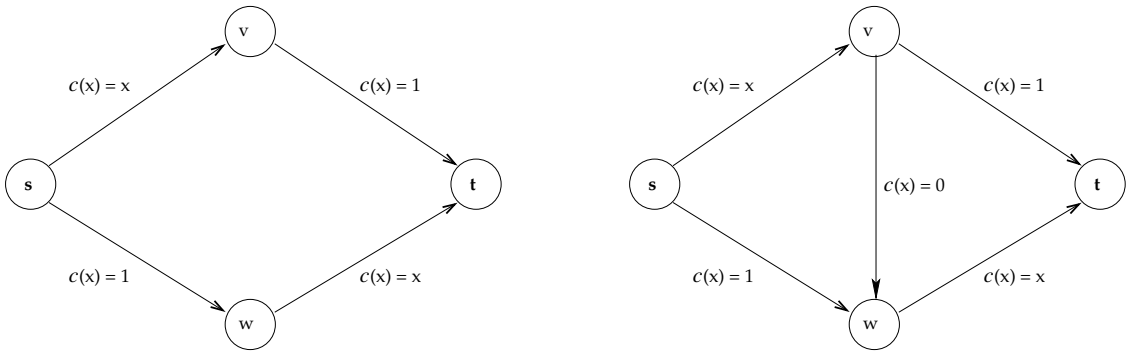
1 Braess's Paradox

The best way to get a feel for selfish routing is through examples. We begin with *Braess's Paradox* (Figure 1) [2]. There is a suburb s, a train station t, and a fixed number of drivers who wish to commute from s to t. For the moment, assume two non-interfering routes from s to t, each comprising one long wide road (with travel time one hour, no matter how much traffic uses it) and one short narrow road (with travel time in hours equal to the fraction of traffic using it) as shown in Figure 1(a). The combined travel time in hours of the two edges on one of these routes is $1 + x$, where x is the fraction of the traffic that uses the route. The routes are therefore identical, and traffic should split evenly between them. (Otherwise, traffic on the more heavily loaded route would have an incentive to switch to the other one.) In this case, all drivers arrive at their destination 90 minutes after their departure from s.

Now, suppose we install a teleportation device allowing drivers to travel instantly from v to w. The new network is shown in Figure 1(b), with the teleporter represented by edge (v, w) with constant cost $c(x) = 0$, independent of the road congestion. How will the drivers react?

We cannot expect the previous traffic pattern to persist in the new network. The travel time along the new route $s \to v \to w \to t$ is never worse than that along the two original paths, and it is strictly less whenever some traffic fails to use it. We therefore expect all

†Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.



(a) Initial network (b) Augmented network

Figure 1: Braess's Paradox. The addition of an intuitively helpful edge can adversely affect all of the traffic.

drivers to deviate to the new route. Because of the ensuing heavy congestion on the edges (s, v) and (w, t), all of these drivers now experience two hours of travel time when driving from s to t. Braess's Paradox thus shows that the intuitively helpful action of adding a new zero-cost link can negatively impact *all* of the traffic![1]

Braess's Paradox shows that selfish routing does not minimize the commute time of the drivers — in the network with the teleportation device, an altruistic dictator could dictate routes to traffic and improve everyone's commute time by 25%. We define the *price of anarchy (POA)* as the ratio between the average commute times in the "selfish" and collectively optimal routings. For the network in Figure 1(b), this is the ratio between 2 and $\frac{3}{2}$ (i.e., $\frac{4}{3}$).

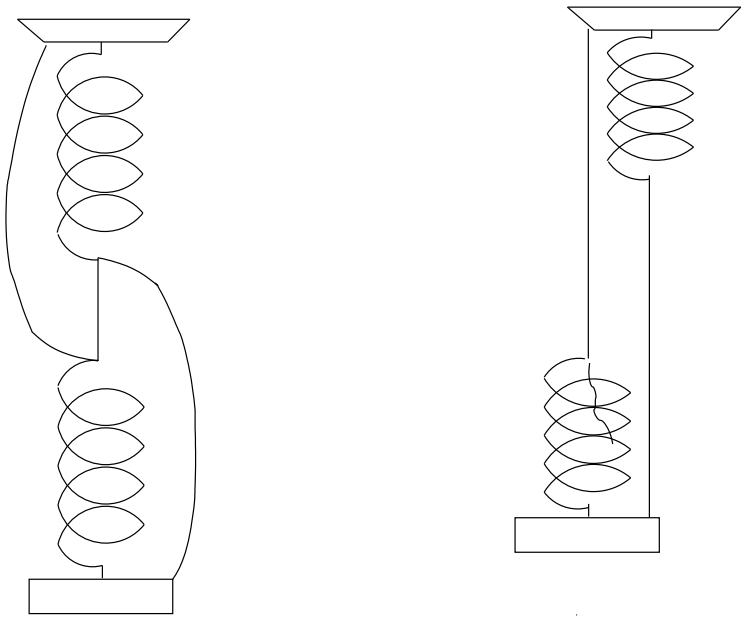
The POA was first defined and studied by computer scientists. Every economist and game theorist knows that equilibria are generally inefficient, but until the 21st century there had been almost no attempts to quantify such inefficiency in different application domains.

Our goal will be to identify conditions under which the POA is guaranteed to be close to 1, and thus selfish behavior leads to a near-optimal outcome and is essentially benign. After we answer this question, we tie the lessons learned into practice. In particular, we'll see a mathematical explanation for the observed fact that over-provisioning of a network leads to good network performance.

1.1 Strings and Springs

As an aside, we note that selfish routing is also relevant in systems that have no explicit notion of traffic whatsoever. Cohen and Horowitz [3] gave the following analogue of Braess's

[1]You might be reminded of the Prisoner's Dilemma; defecting corresponds to taking the zig-zag path, cooperating to one of the two-hop paths. If you've absorbed the Prisoner's Dilemma, then Braess's Paradox is less surprising. After all, if you took away the option of defecting in the Prisoner's Dilemma (akin to removing the edge (v, w)), you would obtain the Pareto optimal solution.



(a) Before (b) After

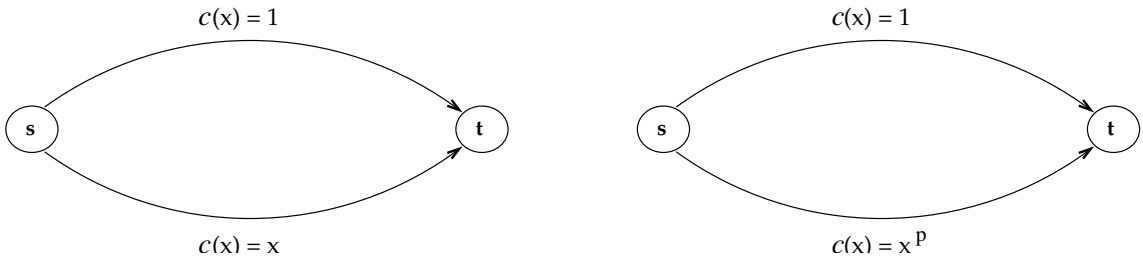
Figure 2: Strings and springs. Severing a taut string lifts a heavy weight.

Paradox in a mechanical network of strings and springs.

In the device pictured in Figure 2, one end of a spring is attached to a fixed support, and the other end to a string. A second identical spring is hung from the free end of the string and carries a heavy weight. Finally, strings are connected, with some slack, from the support to the upper end of the second spring and from the lower end of the first spring to the weight. Assuming that the springs are ideally elastic, the stretched length of a spring is a linear function of the force applied to it. We can therefore view the network of strings and springs as a traffic network, where force corresponds to traffic and physical distance corresponds to cost.

With a suitable choice of string and spring lengths and spring constants, the equilibrium position of this mechanical network is described by Figure 2(a). Perhaps unbelievably, severing the taut string causes the weight to *rise*, as shown in Figure 2(b)! An explanation for this curiosity is as follows. Initially, the two springs are connected in series, and each bears the full weight and is stretched out to great length. After cutting the taut string, the two springs are only connected in parallel. Each spring then carries only half of the weight, and accordingly is stretched to only half of its previous length. The rise in the weight is the same as the improvement in the selfish outcome obtained by deleting the zero-cost edge of Figure 1(b) to obtain the network of Figure 1(a).

This construction is not merely theoretical; on YouTube you can find several physical demonstrations of Braess's Paradox that were performed (for extra credit) by past students in the class CS364A.



(a) Pigou's example (b) A nonlinear variant

Figure 3: Pigou's example and a nonlinear variant. The cost function $c(x)$ describes the cost incurred by users of an edge, as a function of the amount of traffic routed on the edge.

2 Pigou's Example

There is an even simpler selfish routing network in which the POA is $\frac{4}{3}$, first discussed in 1920 by Pigou [6]. In Pigou's example (Figure 3(a)), every driver has a dominant strategy to take the lower link — even when congested with all of the traffic, it is no worse than the alternative. Thus, in equilibrium all drivers use the lower edge and experience travel time 1. Can we do better? Sure — any other solution is better! An altruistic dictator would minimize the average travel time by splitting the traffic equally between the two links. This results in an average travel time of $\frac{3}{4}$, showing that the POA in Pigou's example is also $\frac{4}{3}$.

2.1 Nonlinear Pigou's Example

The POA is $\frac{4}{3}$ in both Braess's Paradox and Pigou's example — not so bad for completely unregulated behavior. Given what we currently know, the coolest thing that could be true would be if the POA of selfish routing was always at most 4/3. (A rather bold guess, given that we've only looked at two examples.) The story is not so rosy in all networks, however. In the nonlinear Pigou's example (Figure 3(b)), we replace the previous cost function $c(x) = x$ of the lower edge with the function $c(x) = x^p$, with p large. The lower edge remains a dominant strategy, and the equilibrium travel time remains 1. What's changed is that the optimal solution is now much better. If we again split the traffic equally between the two links, then the average travel time tends to $\frac{1}{2}$ as $p \to \infty$ — traffic on the bottom edge gets to t nearly instantaneously. We can do even better by routing $(1 - \epsilon)$ traffic on the bottom link, where ϵ tends to 0 as p tends to infinity — almost all of the traffic gets to t with travel time $(1 - \epsilon)^p$, which is close to 0 when p is sufficiently large, and the ϵ fraction of martyrs on the upper edge contribute little to the average travel time. We conclude that the POA in the nonlinear Pigou's example is unbounded as $p \to \infty$.

3 The POA With Linear Cost Functions

Let's again ask the question, what's the coolest thing that could be true? We know that the POA of selfish routing is not always small. Looking back over our three examples, the two examples with POA 4/3 (Braess's paradox and Pigou's example) have different networks but the same kind of cost functions, while the nonlinear Pigou's example has a very simple network but a highly nonlinear cost function. So the coolest thing would be if the POA were small in all selfish routing networks that look like the first two examples, meaning that every edge has a linear cost function (of the form $c(x) = ax + b$, where a, b are nonnegative and can be different for different edges). This may again sound like a wildly optimistic guess (we've only looked at one two-node and one four-node network), but this is in fact true.

Theorem 3.1 ([9]) *In every selfish routing network with linear cost functions, the price of anarchy is at most* 4/3*.*

Theorem 3.1 applies no matter how complex the network topology is, and also for any traffic matrix (with possibly many different origins and destinations). We won't prove Theorem 3.1 here, but the same kinds of arguments are used to prove a different theorem in Appendix A.

4 Network Over-Provisioning

4.1 Motivation

One big advantage in communication networks (compared to transportation networks) is that it's often relatively cheap to add additional capacity to a network. Because of this, a popular strategy to communication network management is to install more capacity than is needed, meaning that the network will generally not be close to fully utilized (see e.g. [5]).

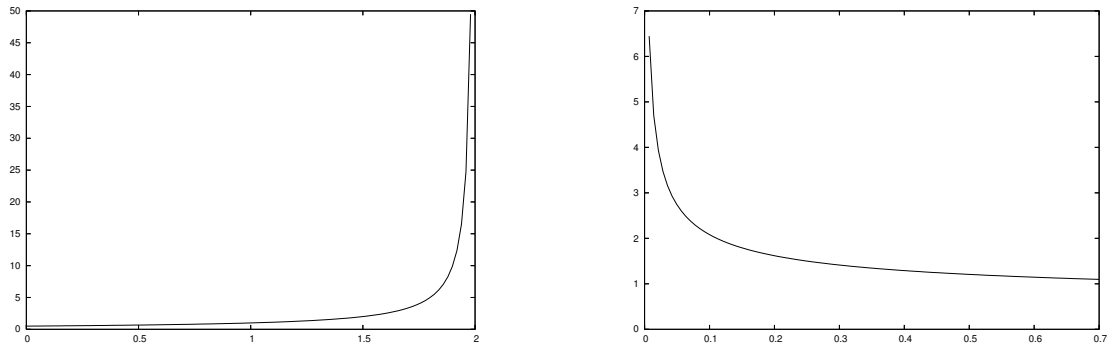
There are several reasons why network over-provisioning is common in communication networks. One reason is to anticipate future growth in demand. Beyond this, it has been observed empirically that networks tend to perform better — for example, suffering fewer packet drops and delays — when they have extra capacity. Network over-provisioning has been used as an alternative to directly enforcing "quality-of-service (QoS)" guarantees (e.g., delay bounds), for example via an admission control protocol that refuses entry to new traffic when too much congestion would result [5].

The goal of this section is develop theory to corroborate the empirical observation that network over-provisioning leads to good performance. Sections 4.2 and 4.3 do this in two different ways.

4.2 POA Bounds for Over-Provisioned Networks

In this section, we consider a network in which every cost function $c_e(x)$ has the form

$$c_e(x) = \begin{cases} \frac{1}{u_e - x} & \text{if } x < u_e \\ +\infty & \text{if } x \geq u_e. \end{cases} \tag{1}$$



(a) M/M/1 delay function

(b) Extra capacity vs. POA curve

Figure 4: Modest overprovisioning guarantees near-optimal routing. The left-hand figure displays the per-unit cost $c(x) = 1/(u - x)$ as a function of the load x for an edge with capacity $u = 2$. The right-hand figure shows the worst-case price of anarchy as a function of the fraction of unused network capacity.

The parameter u_e should be thought of as the capacity of edge e. A cost function of the form (1) is the expected delay in an M/M/1 queue, meaning a queue where jobs arrive according to a Poisson process with rate x and have independent and exponentially distributed services times with mean $1/u_e$. This is generally the first and simplest cost function used to model delays in communication networks (e.g. [1]). Figure 4(a) displays such a function; it stays very flat until the traffic load nears the capacity, at which point the cost rapidly tends to $+\infty$.

We seek a statement of the form "the more over-provisioned a network is, the better its performance." For this, we need a quantitative measure of how over-provisioned a network is. For a parameter $\beta \in (0, 1)$, call a selfish routing network with M/M/1 delay functions *β-over-provisioned* if $f_e \leq (1 - \beta)u_e$ for every edge e, where f is an equilibrium flow. That is, at equilibrium, the maximum link utilization in the network is at most $(1 - \beta) \cdot 100\%$. (So $\beta \approx 0$ is not over-provisioned at all, and $\beta \approx 1$ is wildly over-provisioned.)

Figure 4(a) suggests the following intuition: when β is not too close to 0, the equilibrium flow is not too close to the capacity on any edge, and in this range the edges' cost functions behave roughly like a linear cost function (or at least a low-degree polynomial). Theorem 3.1 gives hope that the POA should be small in networks with such cost functions.

So how can we extend Theorem 3.1 to selfish routing networks with "roughly linear" cost functions? After all, we know that the POA bound of 4/3 is not true in general.

The key idea for a generalization is to rephrase Theorem 3.1 as follows.

Theorem 4.1 *Among all selfish routing networks with linear cost functions, the POA is maximized by Pigou's example.*

Given that Pigou's example has a POA of 4/3, Theorems 3.1 and 4.1 are equivalent. Unlike Theorem 3.1, it's possible to imagine that Theorem 4.1 continues to hold without change for

nonlinear cost functions, for the suitable analog of Pigou's example. This is in fact the case.

Theorem 4.2 (Worst-Case Selfish Routing Networks Are Simple) *[7, 4]] For any set $\mathcal{C}$ of cost functions, among all selfish routing networks with cost functions in $\mathcal{C}$, the worst-case POA is realized by a network with two vertices, two parallel edges, with one edge having a constant cost function.*[2]

For example, taking $\mathcal{C}$ as the set of linear functions, Theorem 4.2 implies Theorem 4.1 and hence Theorem 3.1.[3] We could instead take $\mathcal{C} = \{c(x) = ax^2 + bx + c : a, b, c, \geq\}$ to be set of quadratic functions with nonnegative coefficients, and we'd find the the worst selfish routing network is the same as Pigou's example, except with the cost function $c(x) = x$ replaced by $c(x) = x^2$.

One consequence of Theorem 4.2 is that, for a given set of cost functions, it is usually easy to compute the worst POA that can occur with those cost functions. (Just maximize the POA over all super-simple selfish routing networks.) Carrying out this exercise shows that the POA is reasonably small for cost functions that are low-degree polynomials with nonnegative coefficients (see also Exercise Set #4).

Applying this paradigm to cost functions of the form (1) and β-over-provisioned networks, we can precisely determine the worst-case POA in such networks.

Corollary 4.3 ([8]) *In β-over-provisioned networks, the maximum-possible POA is precisely*

$$\frac{1}{2}\left(1 + \sqrt{\frac{1}{\beta}}\right).$$

Unsurprisingly, the bound above tends to 1 as β tends to 1 and to $+\infty$ as β tends to 0; these are the cases where the cost functions effectively act like constant functions and like very high-degree polynomials, respectively. What's interesting to investigate is intermediate values of β. For example, if $\beta = .1$ — meaning the maximum edge utilization is at most 90% — then the POA is guaranteed to be at most 2.1. In this sense, a little over-provisioning is sufficient for near-optimal selfish routing, corroborating what has been empirically observed by Internet Service Providers.

4.3 A Resource Augmentation Bound

Suppose you have a selfish routing network suffering from poor performance (e.g., because the maximum link utilization at equilibrium is close to 100%). Can we say which of the following two options is better?:

[2]The fine print: $\mathcal{C}$ should satisfy some mild technical conditions, like being closed under multiplication by scalars. Also, it's possible that the worst POA is not achieved in a single network (e.g., if the worst POA is bounded), and rather is approached by the POA in a sequence of simple networks.

[3]Technically, for this one needs to show that among all networks with two nodes, two edges, one constant cost function, and one linear cost function, Pigou's example is the worst. But this is not difficult.

(a) Route traffic centrally. (This may require changing the network routing protocol, for example.)

(b) Upgrade the network, for example by adding additional capacity.

For our last result, we'll prove a sense in which the second option is always the better one. Technically, what we'll prove is a guarantee for selfish routing in arbitrary networks, with no extra assumptions on the cost functions.[4] Given that the worst-case POA is unbounded (Figure 3(b)), what could such a guarantee look like?

In this section, we compare the performance of selfish routing to a "weaker" optimal solution that is forced to send extra traffic. For example, in the nonlinear variant of Pigou's example (Figure 3(b)), the total commute time in the equilibrium is 1. However an optimal solution routes *two* units of traffic through the network, at least one of the edges will have at least one unit of traffic on it, and the commute times suffered by this traffic is at least that by all traffic in the equilibrium.

This "unfair" comparison between two flows at different traffic rates has an equivalent and easier to interpret formulation as a comparison between two solutions that route the same amount of traffic but in operate in networks with with different cost functions. Intuitively, instead of forcing the optimal solution to route additional traffic, we allow the equilibrium to use a "faster" network, with each original cost function $c_e(x)$ replaced by the "faster" function $c_e(\frac{x}{2})/2$.[5] This transformation is particularly easy to interpret for M/M/1 delay functions, since if $c_e(x) = 1/(u_e - x)$, then the "faster" function is $1/(2u_e - x)$ — an edge with double the capacity.[6] The next theorem, after this reformulation, gives a second justification for network over-provisioning: a modest technology upgrade improves performance more than implementing dictatorial control.

Theorem 4.4 ([9]) *For every selfish routing network, the total commute time in an equilibrium with one unit of traffic is at most the total commute time of an optimal routing of two units of traffic.*

Note that this result makes no assumptions about the network topology or about the cost functions (other than the baseline assumptions). We sketch the proof of Theorem 4.4 in Appendix A.

[4] Other than the baseline assumptions that cost functions are continuous, nonnegative, and nonndecreasing.

[5] The equivalence more or less follows from a change of variable.

[6] This result does not follow directly from Theorem 4.2 or Corollary 4.3. It is true that if you keep the routing fixed and double the capacity, then the maximum link utilization is at most 50%. But doubling the capacity of every edge changes the equilibrium, and it is possible that lots of traffic changes routes and floods one or more edges (recall Braess's paradox), resulting in a network that is not β-over-provisioned for β significantly larger than 0.

References

[1] D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice-Hall, 1987. Second Edition, 1991.

[2] D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1968.

[3] J. E. Cohen and P. Horowitz. Paradoxical behaviour of mechanical and electrical networks. *Nature*, 352(8):699–701, 1991.

[4] J. R. Correa, A. S. Schulz, and N. E. Stier Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[5] N. Olifer and V. Olifer. *Computer Networks: Principles, Technologies and Protocols for Network Design*. Wiley, 2005.

[6] A. C. Pigou. *The Economics of Welfare*. Macmillan, 1920.

[7] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

[8] T. Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7):78–86, 2010.

[9] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

A Proof Sketch of Theorem 4.4 (Optional)

For simplicity, assume that the selfish routing network has one origin s and one destination t (the same argument works in general). First, as a thought experiment, consider "freezing" all edge costs at their values at equilibrium. That is, if $\hat{x}_e$ units of traffic use e at equilibrium (with one unit of traffic total), then we're imagining replacing the original cost function $c_e(x)$ of the edge with the constant cost function that always equals $c_e(\hat{x}_e)$. For example, in the nonlinear version of Pigou's example (Figure 3(b)), both edges have cost 1 at equilibrium.

At an equilibrium, all of the s-t paths in use have the same overall delay, call it L, and all paths not in use have overall delay at least L. (Any traffic on a path with overall delay more than L would have an incentive to switch to a faster path.) For example, in the example in Figure 3(b), $L = 1$.

Next, with respect to the frozen edge costs, all paths suffer delay at least L (even when empty), and so the total delay of any routing of two units of traffic suffers total delay at least $2L$ (summing the delays of all 2 units of traffic). This sounds even stronger than what Theorem 4.4 promises—we've proved that the optimal way to route 2 units of traffic suffers total delay at least *twice* that of the equilibrium. The catch, of course, is that we cheated

by using the frozen edge costs rather than the actual cost functions. As a result, we may be overestimating the total delay in the optimal routing of 2 units. Specifically, if the amount x_e^* that the optimal routing sends on edge e is less than the equilibrium amount $\hat{x}_e$, then the frozen edge cost of $c_e(\hat{x}_e)$ might be more than the actual delay $c_e(x_e^*)$ suffered by traffic on this edge in the optimal routing. For example, in Figure 3(b), the frozen (equilibrium) cost of the lower edge is 1, while the delay suffered by traffic on this edge in the optimal routing is only $(1-\epsilon)^d$, which is much less.

What remains to show is the following upper bound on the magnitude of this overestimate:

$$\underbrace{\text{total delay of optimal (frozen costs)}}_{\geq 2L} - \text{total delay of optimal (true costs)} \leq L. \qquad (2)$$

Inequality (2) would imply that the total delay of the optimal routing (with the true cost functions) is at least L, the overall delay of the equilibrium, as claimed by the theorem.

To prove (2), zoom in on a particular edge e of the network. Suppose there are y units of traffic on e in the optimal routing (with 2 units) and z units at equilibrium (with 1 unit). If $y \geq z$, then there cannot be any overestimate—the frozen cost $c_e(z)$ can only be less the the true cost $c_e(y)$. (We're using here that cost functions are nondecreasing.)

So suppose $y < z$. What's the worst that could happen? The frozen cost is $c_e(z)$, and the smallest the true cost $c_e(y)$ could be is 0. (We're using here that cost functions are nonnegative.) This would lead to an overestimate of $c_e(z)$ per unit of traffic. We are assuming that there are less than z units of traffic on e in the optimal routing, so that total overestimate on this edge is no more than $z \cdot c(z)$, the overall delay suffered by the equilibrium on this edge. (Note that this worst case basically happens in Figure 3(b) for large d, on the lower edge, where $z = c(z) = 1$, $y = 1 - \epsilon \approx z$, and $c(y) = (1-\epsilon)^d \approx 0$.) Summing over all of the edges implies that the total overestimate is at most total delay in the equilibrium routing (that is, L), and this establishes (2) and completes the proof.