Mechanisms for Internet Routing: A Study[1]

Aditya Akella[2] Shuchi Chawla[3] Srini Seshan[4]

July, 2002
CMU-CS-02-163

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this paper, we address the issue of Routing in the Internet from a Game Theoretic perspective. We adopt a two-pronged strategy: firstly, we revisit two 'classic' models of the Nash equilibria of a network of selfish flows in the Internet and extend their results for Nash equilibria to what we believe are more realistic settings (for example, we present results for non-linear latency functions). Secondly, we apply our results as well as the 'classic' results for Nash equilibria to designing Routing schemes for networks. The goal of such schemes is not to price network usage but rather to ensure sound overall network performance in the presence of greedy behavior of the participating flows. Finally we show how our results can be employed to build a Wide-Area routing scheme.

[1]This research was sponsored by DARPA under contract F30602-99-1-0518. Additional support was provided by IBM. Views and conclusions contained in this document are those of the authors and should not be interpreted as respresenting the official policies, either expressed or implied, of DARPA, IBM or the U.S. government.

[2]aditya@cs.cmu.edu
[3]shuchi@cs.cmu.edu
[4]srini@cmu.edu

Keywords: Game Theory, Mechanism Design, Load-sensistive Routing

1 Introduction

The last decade has seen a steep surge of interest in Game Theory and it applications, specifically, to the operation of the Internet. Many groups of researchers [1, 3, 4, 7, 9] have shown interesting applications of Game Theory to problems related to the Internet. Issues like Quality of Service, Congestion Control, Multi-Path Routing and Shortest Path Routing, to name only a few, have been analyzed from a Game-Theoretic standpoint.

Of all research pertaining to analyzing the Internet from a Game Theoretic perspective, we are primarily interested in two classes of analyses in this paper. The first class of research studies, spear-headed by the pioneering work of worst-case Nash equilibria by Koutsoupias et al.[2], has aimed at quantifying the effect of greedy behavior in the Internet. The studies in this class[1, 2, 5] have pursued a new paradigm of analysis: one which quantifies the effect of lack of co-ordination among participating entities, as opposed to lack of information (the paradigm of Online-Computation) or computational resources (the paradigm of Approximate Algorithms and Computational Complexity).

A parallel set of research studies, as exemplified by the work of Nisan et al.[3, 4, 6], has concentrated on orchestrating *mechanisms* for the Internet as such. The goal of these studies, broadly, is to come up with pricing or pay-off schemes to end-points/intermediaries in the Internet, so that their greedy behavior is not entirely detrimental to the 'health' of the network. These pricing or pay-off schemes, or mechanisms, are in fact aimed at ensuring that the greedy and selfish actions of the participating entities, given the mechanism, translate into actions that are socially sound.

Though both these sets of studies have provided the research community with tremendous insight into the operation of the Internet, we feel they have not done enough to model the operation of the Internet sufficiently well. Our goal in this paper is to revisit a few of these earlier pieces of work, try to identify how well they relate to the working of the Internet and to extend their results to more realistic and modern settings. In course of achieving this primary goal, we also try to combine ideas and solutions from the first class of studies to those in the second class to arrive at interesting and realistic mechanisms for routing in the Internet.

Specifically, we address the problem of pricing network paths: how should the costs of the various links in a network be set so that selfish agents trying to always select paths providing minimum end-to-end latency pick such paths at Nash equilibrium as would ensure optimum social cost. We consider two definitions of social cost: the average latency in the network and the expected maximum latency of any link in the network. We also assume, as in previous models, the cost of a link to any node is some function of the current load on the link, where the load of a link is defined as the total traffic on the link. The models we consider and the specific definitions are provided in Section 2. Finally, we adapt our mechanisms to constructing a congestion sensitive routing scheme for the Internet. In designing routing mechanisms, our goal is not to extract money from the individual flows. Rather, we aim to make the flows reach optimal social behavior.

A few comments about the analysis we adopt in this paper are in order. Firstly, we would like to point out why our solution concept is that of a Nash equilibrium. We do not analyze Bayes-Nash equilibria in this paper. Though the latter would have been a good choice, it is typically the case that players in the Internet setting cannot be assumed to know anything at all about the competing players, due to the sheer scale of the Internet. Moreover, even if a player had good knowledge of the actions of the others, these would change often with time. In the Internet, it seems natural for anyone to adapt quickly to what the others are doing, irrespective of what one believes they should be doing.

The paper is structured as follows. In Section 2 we elaborate on the abstract model(s) for the

Internet that we consider in this paper. Section 3 discusses the effect of lack of co-operation on the performance of the network with respect to all the models we consider. In Section 4 we discuss Pricing mechanisms for the Internet based on our analysis of the effect of anarchy in Section 3. In Section 5, we show how to construct a feasible dynamic routing algorithm for the wide-area and discuss the applicability of our results to the Internet. Finally, in Section 6 we summarize our results.

2 An abstract model of the Internet

We take a very simplistic view of the Internet. We assume that each flow can be routed over one or many of n available links. The load x on each link is the total flow being routed over it. Each link is governed by a latency function $f(x)$ which is a function of the load on that link and is the cost paid by all individual flows using that link.

The latency of a single link is a feature of the Internet and not in the hands of the mechanism designer. Typically in the Internet, it is of the following form which is derived from the behavior of routers in the Internet using Queueing Theory:

$$f(x) \approx \frac{1}{c-x} \text{ where } c \text{ is the capacity of the link}$$

In the settings that we consider in the rest of the report, such a function is quite hard to analyze. In fact most studies on the subject confine their analysis to only linear latency functions. In parts of the report we will consider linear as well as polynomial latency functions. Polynomial functions are closer approximations to the function given above – both have an unbounded gradient and for loads not approaching capacity of a link, polynomials are representative of true latencies.

Now three questions remain:

- Who are the agents?
- What strategies can they use?
- What is the social cost function?

The first two questions correspond to how much freedom a user has in deciding the flow on each link. We consider three types of strategies in this report.

S.1 Each single packet is an agent. It alone cannot make a significant impact on the flow on any link. It chooses a link that has minimum latency, and in case of a tie, chooses arbitrarily. This model would be representative of an internet in which flows are infinitesimally divisible.

S.2 Each flow of 1 unit is an agent. It chooses a link of minimum latency (breaking ties arbitrarily) and routes its entire flow along that link. This is a variant of S.1 for which flows are unsplitable.

S.3 Once again each flow of 1 unit is an agent. It chooses a link with some probability and routes its entire flow along that link. This can be thought of as mixed strategies as opposed to only pure strategies in S.2.

Note here that whenever we say that the link with minimum latency is chosen, we imply that the user is comparing latencies including his flow along the link that he is considering.

The second question asks what is the function that a central planner wants to minimize in the Internet. We consider two ways of combining latency functions on links to get a social cost function:

F.1 a weighted average of latencies on every link, weighted by the total amount of flow on each link.

F.2 maximum latency over all links.

We do not wish to argue which of these scenarios applies most to the Internet. We will analyze all of the above. First we turn our attention to the behavior of Nash equilibria in the Internet.

3 How bad is Anarchy?

Before designing a mechanism, it is important to motivate that one is necessarily required. In the case of the Internet, in the absence of a pricing mechanism, users selfishly route so as to minimize their own latency. We call this situation and the resulting value of the social cost Nash equilibrium and Nash cost respectively. If Nash cost is not much worse than the optimal social cost of the system, then we need not employ the additional computational overhead of designing a pricing mechanism. In this section we will discuss how different Nash cost can be from the optimal cost in the worst case for various models of the Internet. For measuring performance, we will consider the ratio of Nash cost to the optimal cost. This ratio is sometimes known as the Coordination Ratio.

3.1 Minimizing Average Latency for Pure Strategies

The question of how bad a Nash cost is has been studied by many people before. Roughgarden et al[5] study the question for (S.1,F.1). They show that if the latency functions are linear, then, Nash cost is at most 4/3 times the optimal cost. However, for arbitrary latency functions, they give only a bicriteria result which says that Nash cost cannot be worse than optimal cost at half the capacity.

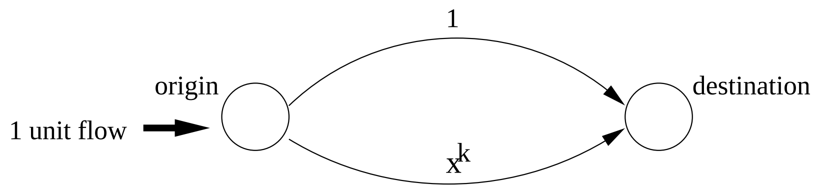

Figure 1: A simple example of bad coordination ratio when latency gradient is unbounded

It is easy to see that for arbitrary latency functions (in fact specifically for functions with unbounded gradient), the coordination ratio cannot be bounded. A simple example is given in figure 1. In this example, in Nash equilibrium, all packets go to link 2, since the cost of that link is always less than 1, irrespective of the amount of flow on it. On the other hand, the optimal solution here is to route a small amount of flow on link 1, causing the cost of link 2 to go down to $1 - k(K+1)^{-(k+1)/k}$ which tends to 0 as k increases. Hence the coordination ratio can be made arbitrarily large. The bicriteria result says that the Nash cost will always be less than the optimal cost if Opt had to route 2 units of flow in the same network.

The bicriteria result can be interpreted as saying that we can help the network by doubling its capacity. However, this is not a practical solution for the Internet as doubling the capacity would imply that agents simply increase their flow forgoing all benefits.

[5] give a similar bicriteria result for minimizing latencies in which flow is unsplitable and agents can use pure strategies. In section3.3 we will discuss the situation of minimizing average latencies when agents can use mixed strategies.

3.2 Minimizing Maximum Latency

Koutsoupias et al[2] studied Nash equilibria in the scenario of minimizing maximum latencies when agents use mixed strategies (S.3,F.2). They show that if there are only two links in the system, Nash cost is at most 3/2 times worse than the optimal cost. For m links, this value increases to $\frac{\log m}{\log \log m}$. Both these results hold only for linear latency functions.

We will now try to put the results of [2] in perspective and will then extend it to other cases. We begin with an instructive example of 2 links and square latencies.

Definition 3.1 *The support of an agent i is the set of links S_i for which the probability of routing its flow on any link belonging to that set is non zero for agent i.*

Observation 3.1 *The expected latency seen by agent i on any link belonging to his support is the same and less than that on any other link.*

Based on the above observation, we can derive the Nash equilibrium for the case of two links. Assume there are k agents and both the links are governed by the latency function $f(x) = x^2$. Further assume that at Nash equilibrium, a of the agents have only link 1 in their support, b of the agents have only link 2 on their support and the remaining c agents use link 1 with probability p_i. Let these sets of agents be A, B and C respectively.

Lemma 3.2 *For all agents $i, j \in C$, $p_i = p_j = p^*$.*

Due to lack of space and to preserve continuity, we will not prove this lemma rigorously, however we give an intuitive argument here. Firstly notice that the equations defining the equilibrium (one for each agent, saying his strategy is the best) are symmetric in the probabilities p_i. Consider two agents i and j with their respective probabilities related as $p_i < p_j$. Let c_i^1 and c_i^2 be the expected costs for links 1 and 2 seen by the agent i. Since the support of agent i includes both the links, we have, $c_i^1 = c_i^2$. Now, consider the situation from the perspective of agent j. His view of the situation is identical from that of agent i except that his counterpart routes on link 1 with probability lesser than his own. So, $c_j^1 < c_i^1$. Similarly, we can argue that $c_j^2 > c_i^2$. This implies that $c_j^1 < c_j^2$. So the support of j cannot contain the link 2 and we arrive at a contradiction.

With this observation, we now try to find this probability p^*. For every agent, the equation defining Nash equilibrium is given by the following:

$$\begin{array}{lrcl}
\text{Agents in } A: & \sum_{i=0}^{c} p^i(1-p)^{c-i}\binom{c}{i}(i+a)^2 & < & \sum_{i=0}^{c} p^i(1-p)^{c-i}\binom{c}{i}(c-i+1+b)^2 \\
\text{Agents in } B: & \sum_{i=0}^{c} p^i(1-p)^{c-i}\binom{c}{i}(i+1+a)^2 & > & \sum_{i=0}^{c} p^i(1-p)^{c-i}\binom{c}{i}(c-i+b)^2 \\
\text{Agents in } C: & \sum_{i=0}^{c} p^i(1-p)^{c-1-i}\binom{c-1}{i}(i+1+a)^2 & = & \sum_{i=0}^{c} p^i(1-p)^{c-1-i}\binom{c-1}{i}(c-i+b)^2
\end{array}$$

Simplifying the binomial expressions and solving the equations together, we get, $p^* = \frac{1}{2} + \frac{b-a}{2(c-1)}$.

Now, the optimal allocation for this scenario is to route half of the flows over link 1 and the rest along link 2. This gives us optimal cost $\lceil \frac{a+b+c}{2} \rceil^2$.

On the other hand, cost of the Nash equilibrium computed above is given by:

$$\sum_{i=0}^{i_0} p^i(1-p)^{c-i}\binom{c}{i}(c-i+b)^2 + \sum_{i=i_0+1}^{c} p^i(1-p)^{c-i}\binom{c}{i}(i+a)^2 \qquad \text{where } i_0 = \lfloor \frac{b-a+c}{2} \rfloor$$

Unfortunately, simplifying this expression in terms of a, b and c is quite hard as the binomial expressions above are incomplete. In fact it is also difficult to give a good upper bound to the above expression. However, we can simplify it for a special case.

When a and b are zero, and $p^* = 1/2$ for every agent, the social cost turns out to be $\frac{1}{2^{n-1}} \sum_{i=n/2}^{i=n} \binom{c}{i} i^2$ which is less than $\frac{3}{4}n(n-1)$. This is at most 3 times worse than the optimal cost which is $n^2/4$ as before.

This is not a very interesting observation by itself, as we have only bounded the cost of one Nash equilibrium. But we believe that in the given situation, this is the worst Nash equilibrium. In general, the larger the number of agents that want to route probabilistically, the more is the chance that some times the flow will be highly uneven and that results in a large value for expected maximum latency.

Conjecture 3.3 *The worst Nash equilibrium for the situation of minimizing maximum latency when agents use mixed strategies and there are 2 links both having square latencies is the one in which all agents have a probability of* $1/2$ *to route their flow on the 1st link. In this case, the Nash cost is less than* 3 *times worse than the optimal cost.*

Next we consider the case of pure strategies, again while minimizing maximum latency.

Consider the case where each agent controls a single packet and tries to route it on the link with the minimum latency (scenario S.1). Clearly, in this case, in Nash equilibrium, every link with non zero flow on it will have equal latency. This is because if to the contrary, there exist two links with unequal latency, then some ϵ amount of flow on the higher latency link would be able to reduce its latency by shifting to the lower latency link.

On the other hand, consider the optimal solution when the central planner wants to minimize the maximum latency function (scenario F.2). In this case again, all links in the optimal solution will have equal flow, otherwise, the central planner would be able to shift some flow from a higher latency link to a lower latency link reducing the maximum.

So we see that the Nash equilibrium and social optimum for this situation are the same!!

Going from S.1 to S.2, we can use a similar argument, where now both the optimal and Nash equilibria can only route integral flows.

Theorem 3.4 *Nash cost and social cost are equal for the situation of minimizing maximum latency when agents are restricted to pure strategies.*

3.3 Minimizing Average Latency for Mixed Strategies

Finally, we consider the situation of minimizing average latency where agents are allowed to use mixed strategies (S.3,F.1). We are able to analyze this situation when all links are governed by the same latency function.

Firstly, notice that if we compare this situation to that of minimizing maximum latency with mixed strategies (S.3,F.2), we are only changing the social cost, but the incentives to agents, and hence their behavior remains the same. So the Nash equilibrium in both the cases will be the same. Since the average of a bunch of numbers is at most their maximum, the nash cost for (S.3,F.1) is upper bounded by the nash cost for (S.3,F.2).

Similarly, consider the optimal solution. If we assume that the gradients of latency functions are increasing (which is a reasonable assumption for the Internet), then whenever there are two links with unequal latency, moving some amount of flow from the greater loaded link to the lesser loaded link will reduce the gradient of latency on the greater loaded link more than the increase in the gradient on the lower link. So, this will decrease the overall social cost. This suggests that for F.1, the optimal solution in this case would be to distribute loads to links equally (such that latencies of the two are equal). This is the same as the optimal solution for F.2. Interestingly, since the latency on every link is the same, the optimal cost for (S.3,F.1) is equal to the optimal cost for (S.3,F.2).

Thus the ratio of nash cost to optimal cost for (S.3,F.1) is at most the ratio for (S.3,F.2) and the same bound of $\frac{\log m}{\log \log m}$ for m links holds.

More on Mixed strategies and a lower bound

Combining the results in this section so far, we notice that whenever agents are allowed to use mixed strategies, the situation becomes much worse. (Please see table1)

	Pure strategies	Mixed strategies
Avg latency	4/3	$\frac{\log m}{\log \log m}$
Max latency	1	$\frac{\log m}{\log \log m}$

Table 1: Coordination ratios for linear latency functions

In order to understand this better, consider a situation in which all agents have a support of size greater than 1. Let C denote the set of costs obtained from sending each agent's flow to any one of the links in its support. Then the cost of this equilibrium is some convex combination of costs in C (by definition). Now, let c^* denote the minimum cost among these costs. Notice that given any strategies and probabilistic choices of agents, (1) there is zero probability for the cost being less than c^*, and (2) with positive probability cost is greater than c^*. Clearly for any set of supports then, we can find alternate pure strategies for agents that give a lesser social cost.

Based on this discussion we can safely conjecture that the worst Nash equilibrium for minimizing maximum latency is for all agents to distribute their probabilities equally among all links.

For the situation of m flows on m links, this observation easily gives us a bound on the coordination ratio. This situation is similar to asking: suppose we throw m balls into m equally likely bins, what is the maximum expected number of balls in a bin. The answer $\frac{\log m}{\log \log m}$ is a well known mathematical result and leads to the corresponding result given in [2].

4 Pricing Mechanisms

In the last section we discussed upper bounds on the Nash cost in terms of the optimal cost for various models. We found that in many situations, Nash cost can be arbitrarily worse than the optimal cost. In those scenarios, we would like to device mechanisms which lead to an equilibrium which is reasonably close to the social optimum.

In this section we will present a mechanism that minimizes average latency of the network. We will first analyze that mechanism for the case of packets as agents (S.1). Later we will discuss under what conditions the mechanism can be generalized to the case of flows as agents (S.2). In section 5 we will adapt this mechanism to the Internet and discuss implementation issues.

In this situation of minimizing average latency when packets are agents, the social cost function is given by $\sum_j x_j f(x_j)$, where x_j is the total load on link j and f is the latency function for that link. This quantity is equal to $\sum_i x(i)l(i)$, where $x(i)$ is the amount of flow of agent i and $l(i)$ is the latency seen by that agent. Notice that this is a utilitarian function. This suggests that we should be able to use the Vickrey-Clarke-Groves mechanisms here. We will motivate the same mechanism through a different argument in this section.

We first make two observations based on our discussion in the last section.

Observation 4.1 *At Nash equilibrium, all links have the same latency.*

Observation 4.2 *In the social optimum, all links have the same latency gradient.*

The above two observations immediately suggest the following mechanism: Set shadow prices along a link to be the gradient of the function $xf(x)$ where x is the flow along that link and f is the latency function governing it.

This result holds for arbitrary latency functions (provided that they are differentiable), and even if the agents have some utilities that are functions of the latency that they see and their preferences are quasi-linear.

Notice that the observations above can be extended with some degree of approximation to the case when flows are unsplitable, but granularity is sufficiently high. By granularity we mean that the amount of flow carried by a single agent is a small fraction of the total amount of flow and so, decisions of a single agent alone cannot effect the latency on any link or social cost considerably. The amount by which social cost can be effected by a single agent governs how well our mechanism will perform in this situation. Notice here that the bounds on coordination ratio given by [5] also depend on this granularity. In the next section we will discuss the case of the Internet and will show that in the Internet granularity is in fact considerably high.

Now we come to the issue of actual implementation. The shadow prices given by $(xf(x))' = f(x) + f'(x)$ can easily be computed by routers in the Internet, as the current latency and its gradient is known to the routers. Given this, the routers can now use the *Congestion Notification bit* in TCP protocol to send guidance regarding shadow prices to end-hosts or agents. The CN bit has been used in the past for congestion control in the Internet and has been know to be quite effective. Our protocol is based on existing features and does not require much extra functionality. This we believe is a strong positive feature of our mechanism.

We still need to argue, that the above mechanism achieves social optimum even in the presence of Internet traffic which is not entirely like the simple cases that we have considered so far. We present an argument in the following section.

5 Building a Robust Wide-Area Routing Scheme: Implementation Issues

Many issues need to be considered when extending our solution into a viable Internet routing protocol. Firstly, flows in the Internet use TCP as a transport mechanism, which cannot tolerate reordered packets. As a result, a solution for the wide area cannot make any assumption about the splitability of flows. Secondly, flows in the Internet carry varying amounts of packets and hence offer different amounts of load on the network. Studies have shown that flows in the Internet are mostly short-lived (carry a few packets) while a few of them are long-lived (carry a lot of packets). However, most data in the Internet is carried in the long-lived flows. What this means in terms of the models we have considered in this paper, is that we cannot assume that the total flow of all participating flows is identical. At best, a bimodal distribution could be a reasonable approximation, but a solution that is independent of the total amount of load due to any flow is more desirable.

In this section, we show via informal arguments, how the statistics of flow life time and arrival rate in the wide-area can be used to construct a routing protocol. Our work still leaves many questions unanswered, though. We would like to verify that some of the assumptions made by the following discussion are in fact true for the Internet. Moreover we would like to observe how fast the system converges to an equilibrium.

5.1 Nash Equilibria Determined by Short-Lived Flows

The pricing solution of Section 4 works for splitable flows. This scheme can be easily extended to unsplitable but short lived flows to some degree of approximation (we don't have a formal proof for this; the arguments are only intuitive). The essential idea is as follows: if all the selfish flows were short lived, and unsplitable, then one can 'replace' each agent in the previous solution (which was a single packet) with a short lived flow and the Nash equilibrium has the same characterization as in the infinitely splitable case and hence the same pricing scheme as the one where each packet is a selfish agent should work here too. This is because as in S.1, individual flows will be quite insignificant compared to the total flow in this situation.

In the setting described in [5], if each agent chooses to pick the path with minimum end to end latency (that is each agent uses the end to end latency as the cost of the path), then starting from an 'empty' network or a network that is not at equilibrium, packets would route themselves greedily, so that ultimately all the potential paths for any flow look similar: at this stage Nash equilibrium is attained, and once attained the equilibrium stays. Now, we talk about the nature of the equilibrium here as we want to discuss how bringing in long lived flows changes our assumption.

Here is what we propose for any flow which is unsplitable: the first packet of the flow picks the least cost path and all the remaining packets on the flow follow this path (as in S.2).

5.2 The Actual Working and the Impact of Long-Lived Flows

Imagine a network of short lived flows, routed as above, at equilibrium. Now, assume a long lived flow arrives and the first packet on this flow chooses a path. This long lived flow disturbs the load on the links in the network and as a result the equilibrium of the short lived flows is disturbed. However, the currently existing short lived flows would end, more would come in, and these new short lived flows would bring about a new equilibrium. Thus the short flows come in and out of equilibrium with each arriving and departing long lived flow. But all the while, since we are routing the short lived flows according to pricing scheme, they will contribute to reducing the average end to end latency.

Our concern now is with long lived flows. The packets in the long lived flows will start contributing to minimizing the end to end latency (will be on an optimal path) , as soon as the short lived flows have settled to the equilibrium as until then the packets in the long lived flows will be taking paths that are not optimal according to our pricing scheme. Hence it is important to ensure or satisfy ourselves that convergence is quick. Now, convergence is ensured by short lived flows arriving at quick rates (it is not dependent on the currently active short lived flows having to re-route their packets).

5.3 Other Implementation Issues

In the future, we would like to make sure using measurements of flow arrivals at a BGP gateway that this is actually true. As an alternative, we are thinking of deriving the optimal arrival rate to achieve a given time-span of instability. We could also use other tricks like temporarily make some links unavailable to some flows to ensure that the flows would now route so that the new arrival rate is what we want. There are other issues like, how often should we send notifications regarding shadow prices (updates) to agents, do we want to apply this solution to backbone networks or peering points, what is the right update interval which still need to be addressed.

Also, there are other points worth mentioning here. Firstly, from the point of view of a short lived flow, all a long lived flow does is to change the latency functions of the links that it is taking. Hence, this is like a situation in which the network would change its functions by some amount after some reasonably long period. So at that point, the flows would have to form a new equilibrium. Another point is that if we think of the core as a single node, then every short lived flow essentially has only a few paths to choose from and so does not have to worry about too many long lived flows. Also, not only do we need short lived flows need to arrive at a sufficiently fast rate, but long lived flows should arrive much less frequently. However, we think this will not be an issue at peering points if we aggregate simultaneous long lived flows in one group and use this group as an abstraction for a long flow.

5.4 Current Status of Implementation

We are collecting traces from Akamai to study the arrival patterns of short and long-lived flows to confirm our assumptions. We have also built a simple multi-path routing scheme in Network Simulator (NS-2). Due to lack of time, we could not take any measurements in the simulator. In the future, we would like to extend this solution to one providing Quality of Service: different flows are provided different levels of service by the network depending on the price paid for usage.

6 Conclusions

In this project we considered the problem of routing in the Internet. Our goal is to minimize average or expected maximum latency in the Internet. We observed that in the absence of a mechanism, a Nash equilibrium can be arbitrarily worse than the social optimum. In order to improve the situation, we designed a mechanism which provably leads to the social optimum when flows are splitable. We argued that the same holds when most flows are sufficiently small, which is the case in the Internet. We also showed how to implement this mechanism in a distributed manner in the Internet, without using any additional functionality than what is already present.

As future work, we would like to test out our routing mechanism using the NS simulator and would like to prove some of our claims rigorously.

Moreover, even though we studied Nash equilibria for the situation of minimizing expected maximum latency when agents are free to use mixed strategies, we were unable to design a suitable mechanism for those cases. We would like to work some more on that too.

References

[1] P. Key, D. McAuley, P. Barham, and K. Laevens, *Congestion Pricing for Congestion Avoidance.* Microsoft Research Technical Report MSR-TR-99-15, Feb 1999.

[2] Elias Koutsoupias and Christos Papadimitriou, *Worst-case equilibria.* Symposium on Theoretical Aspects of Computer Science, March 1999.

[3] N. Nisan and A. Ronen, *Algorithmic mechanism design.* Symposium on Theory of Computation , 1999.

[4] N. Nisan, *Algorithms for selfish agents*. Symposium on Theoretical Aspects of Computer Science, March 1999, p. 1–15.

[5] T. Roughgarden and E. Tardos, *How Bad is Selfish Routing?*. Symposium on the Foundations of Computer Science, 2000, pages 93–102.

[6] T. Roughgarden, *Stackelberg Scheduling Strategies*. Symposium on Theory Of Computation 2001, pages 104–113.

[7] H. Moulin and S. Shenker, *Strategyproof Sharing of Submodular Costs: Budget Balance Versus Efficiency*. To appear in Economic Theory. Preprint version at `http://www.aciri.org/shenker/cost.ps`

[8] Scott Shenker, *Service Models and Pricing Policies for an Integrated Services Internet*. In Public Access to the Internet (B. Kahin and J. Keller, eds.), pp. 315-337, MIT Press, 1995.

[9] K. Jain and V. Vazirani, *Applications of Approximation Algorithms to Cooperative Games*. Symposium on Theory of Computation, 2001.