# TRECON: A Framework for Enforcing Trusted ISP Peering

Zhengqiang Liang
Department of Computer Science
Wayne State University
Email: sean@wayne.edu

Weisong Shi
Department of Computer Science
Wayne State University
Email: weisong@wayne.edu

***Abstract*— We envision that neglecting economic factors and trustworthiness evaluation of ISPs is one of the obstacles to developing next generation Internet (NGI). In this paper, we take the initial step to build a general framework called TRECON, which combines an adaptive personalized trust model (*a*PET) with an economic-based approach and provides independent routing among ISPs. With TRECON, autonomous organizations (e.g., ISPs) with varied interests and optimization criteria are smoothly integrated together to achieve better scalability, isolation and self-management. The evaluation results show that the trust-based strategy (*TRU*) performs much better than the global shortest path routing (*SPA*) approach in terms of delay, reliability and economic incentives.**

## I. Introduction

In the current Internet, BGPv4 [12] is the underlying routing protocol that is used by ISPs to implement their peering policies and control the traffic exchanged at the peering points. However, there are three types of deficiencies of current BGP [2], [9], [14]. First, BGP has no requirements on the routing structure, which improves the general applicability, but also brings a severe oversized routing table problem. Since BGP reveals complete AS (autonomous systems) path information, the increase of the number of ASes can lead to the exponential increase of the routing table. Second, the path-vector routing in BGP also makes the local routing event globally visible, which leads to bad scalability and poor fault isolation properties. Third, while revealing complete path information, BGP keeps policy information private, which hinders the cooperation among ISPs and the discovery of routing with specified routing quality. In addition to the routing issues in current ISP peering, another issue ignored by most researchers is, *there are no mechanisms to support the economic ecology in ISP peering*. The report on a recent NSF workshop [10] reveals that the future design for Internet must take competition and economic incentives into account; ignoring the economic ecology in ISP peering is inevitable to reduce the incentives for the ISP to provide better services and cooperate with other ISPs. Hence it is difficult to motivate users to seek services from ISPs. The solution to NGI must solve the incentive problem for both ISPs and users.

Market-based economics [13] is promising to support service differentiation. In this paper, we envision that combining the trustworthiness evaluation of neighbors and the economic approach makes the previously separable concerns – *incentive compatibility* and *computational tractability* – be jointly addressed. There are three requirements to build the effective ISP peering:

**(1) *Effective Routing Structure*:** Building effective routing structure is fundamental to build effective routing among ISPs. A good routing structure can alleviate the pressure of routing protocol design and routing table storage.
**(2) *Intelligent Routing Decision*:** Intelligent routing decision helps improve the network performance, reduce costs, increase reliability, and support flexible policies.
**(3) *Healthy Incentives-based Environment*:** There are incentive factors behind ISP peering. Every ISP aims to get economic benefits by attracting as more as possible users. Users need good services from good ISPs. A healthy environment of ISP peering should direct requests of users to the good ISPs, so as to maximally meet the demands of both ISPs and users at the same time.

In this paper, we propose **TRECON**, a **TR**ust-based **ECON**omic framework to enforce the next generation trusted ISP peering. TRECON consists of three components: *ISP clustering, adaptive personalized trust model*, and *trust-based economic model*. Because of its flexibility and adaptability, TRECON can be easily adapt to the general network routing, such as routing in sensor network. Since the trustworthiness is related to service quality, TRECON also has potential to support QoS-based routing, which is one of the requirements of NGI [10].

We build our simulation using NetLogo [16] to evaluate our framework. We compare the trust-based routing selection (*TRU*) with the global shortest routing (*SPA*) approach, a globally optimized approach, in the same network topology. We found that *TRU* has better performance than *SPA*. The results indicate that, using trustworthiness information to direct the routing is promising for the design of NGI. The contributions in this paper include three-fold: (1) We build an adaptive trust model *a*PET to evaluate the quality of routing for neighbors; (2) Based on *a*PET, we propose an economic model to solve the economic incentives in ISP peering; and (3) We evaluate the routing performance of TRECON from the perspectives of delay and reliability of routing, and the economic incentives of ISP peering.

## II. The TRECON Framework

The TRECON framework introduces efficiency, incentives, and profits to ISP peering by combining the trust inference and the economic-based approach to . There are three main components in the TRECON framework. A clustering algorithm is first introduced to partition the ISP network into different clusters. An adaptive trust model *a*PET is then applied to direct

the intra-cluster routing. Finally, an simple economic approach is employed for ISP resources sharing.

### A. The Clustering Algorithm

The rapid growth of ISPs makes the routing among ISPs more and more complicated. Path-vector routing is used in BGP [12], which can avoid loops in routing. However, it also causes a severe problem that the routing table becomes oversized. We propose to build a hybrid routing structure, similar to HLP [14], by combining the advantages of path-vector and link-state protocols. HLP works at the AS level, while TRECON works at the ISP level. Our basic idea is to build another layer of routing structure to change the granularity of routes and to extract the stable route from the network. After clustering, the routing is separated into two parts, *inter-cluster routing* and *intra-cluster routing*. The inter-cluster routing is path-vector based, and globally fixed after the clusters have been formed. All ISPs need to store this global inter-cluster routing table. Since the routing unit is based on clusters, the inter-cluster routing table is far more smaller than the routing table in BGP. For the intra-cluster routing, each ISP locally decides the next hop within one cluster mainly based on the trustworthiness of neighbors. Since the routing decision is not globally visible, the routing failure and update then can be limited to one cluster. So under the cluster structure the routing has good isolation and fault tolerance properties. Comparing to the traditional BGP, our cluster structure is more scalable because each ISP just needs to store the small global inter-cluster routing table and the trustworthiness information about the very limited neighbors.

The basic idea of the clustering algorithm is to find out the bridges in the ISP network first, then using the bridges to separate the ISPs into different clusters with certain size. After clustered, the ISP clusters form a line or a circle, so that the inter-cluster routing is unique. Not all graphs are clusterable, especially for the graph with high connections and no bridges. We can find out the substructures which are clusterable by applying the clustering algorithm and then form a hierarchical structure. Also, we may carefully remove some non-bridge links within the whole ISP network so as to make it clusterable. Due to the space limit, we leave the details of the clustering algorithm in our technical report [8].

### B. a*PET:* a*daptive PErsonalized Trust Model*

After the network has been clustered, the inter-cluster routing is fixed. What we need to solve next is the intra-cluster routing. In TRECON the intra-cluster routing is mainly directed by the trustworthiness information, which is provided by a novel *a*daptive **PE**rsonalized **T**rust model called *a*PET. In this section, we present the details of the *a*PET model, which is intended to be applied to any open distributed environment.

In the following sections, the entity which provides the service to the others is called the service provider (*SP*). The entity sending out the rating is called the rater. In *a*PET, the trust is defined as *the subjective probability by which an individual A expects that another individual B performs a given action as good as expected in a certain time*. Actually *a*PET is built on our previous PET model [6], a personalized trust model proposed in the context of P2P systems, and our thorough analysis of the state-of-the-art rating aggregation algorithms [7]. Different from *PET*, *a*PET is able to adaptively change the weight for trustworthiness derivation according to the change of the environment. In ISP peering, the trustworthiness information of neighbors provided by *a*PET is used to help find a route with "good" quality, depending on the specific requirements of applications. Several key observations from our previous work are summarized as follows. These are the design principles of the *a*PET model :

**Pr.1** *Ratings (opinions of other SPs assessing the quality of others) are not always as helpful as what we expected, especially when the system is facing bad raters and highly dynamic behaviors.*

**Pr.2** *In certain circumstances, the simple rating aggregation algorithm such as the average aggregation algorithm performs better than the complicated ones, especially when there are considerable number of bad raters in the system.*

**Pr.3** *When the environment is getting worse (many bad raters, bad and dynamic* SP*s), lowering the weight of ratings and increasing the size of the neighbor set are very helpful to improve the performance of the trust model.*

Besides, we also abstract two high level requirements of trusted models in open environments:

**Re.1** *The weight of the information to derive the trustworthiness should be adaptive to different situations, especially under a severe environment.*

**Re.2** *The trust model should not only be able to promptly find out the fixed bad* SP*s, but also to catch the suddenly spoiled* SP*s and be sensitive to the strategic oscillating* SP*s*.

In *a*PET, the trustworthiness `T` is derived as $T = W * R + (1-W) * I$, where `I` is the interaction-derived information and `R` is the rating. `I` is achieved through direct experience with other *SP*s, which is regarded as a kind of reliable source for the derivation of the trustworthiness value. However, relying on only this kind of information is not efficient enough to derive the trustworthiness value. `R` is introduced for the efficiency purposes, which helps the system discover the quality of other *SP*s even without direct experiences. But rating is not reliable due to the dynamics of the environment and malicious raters. Integrating these two kinds of information is promising by inheriting their corresponding advantages while inhibiting their disadvantages. Note that the combination of these two factors has been proposed for a while, however, the novelty of *a*PET is the adaptiveness of weights, as discussed below.

Figure 1 shows an overview of *a*PET. Each *SP* has its own neighbor set, as illustrated on the left side. The neighbor meeting the requirements is selected by following certain rules for cooperation when needed. This special neighbor is called the *cooperator*. Bad neighbors may be purged from the neighbor set to the blacklist. New neighbors are chosen from the stranger set when the neighbor set becomes small. The neighbor set is stored in the neighbor list, which is a global data structure in *a*PET. In Figure 1, there are three

neighbors, *SP* A, B, and C. Correspondingly there are three elements in the neighbor list, each of which includes the fields of entity ID, the trustworthiness value, and the ripple level number (The ripple model is an optional model in addition to *a*PET to improve the ability of *a*PET to resist the dynamic *SP*s. Its details can be seen in our technical report [8].) For every neighbor, two local data structures, a rating queue and a history table, are used to store the rating and interact-derived information respectively. To meet the requirement of **Re.1**, another global data structure, an environment alert queue, is employed to sense the severity of the environment. The neighbor list, the rating queue, the history table, and the environment alert queue are all FIFO queues. Their sizes are denoted as $S_N$, $S_R$, $S_H$, and $S_E$ respectively. As described in

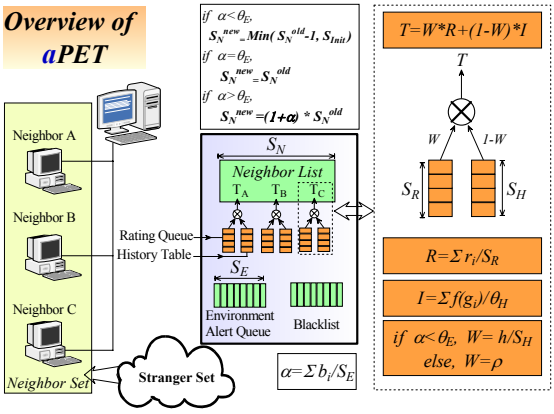



Fig. 1. The overview of the *a*PET model.

**Pr.2**, we find that paying much energy in the rating aggregation algorithm is not wise considering the cost and the payback [7]. In *a*PET, the simple average scheme is used to aggregate the ratings. The rating $r_i$ is the $i^{th}$ element in the rating queue, which can be either 0 (bad) or 1 (good). The interaction-derived information `I` can be obtained from the feedbacks of agents or the self-observation of *SP*. For a specific neighbor, its `I` is calculated as the average score for a specific time span: $(\sum f(g_i))/\theta_H$, where $g_i$ is the $i^{th}$ observed behavior, $f$ is the behavior evaluation function which maps the behavior to a score (in our current *a*PET model $f(g_i) = 1$ if $g_i$ is good behavior, otherwise $f(g_i) = 0$), and $\theta_H$ is the threshold (its value is set to 10 in the simulation).

The adaptiveness of *a*PET mainly embodies in its capability to self adjust the weight *W* and the size of the neighbor list $S_N$ according to the severity of the environment, which is designed based on **Pr.3**. There is one important metric, the environment-aware factor $\alpha$, to guide the adaptiveness. The environment alert queue is used to sense any change in the surrounding environment. It records the quality of the most recent received services. Similar to the rating, the $i^{th}$ element $b_i$ in this queue can be either 0 (bad service) or 1 (good service). $\alpha$ is defined as the proportion of bad services in the most recent interval. A large $\alpha$ indicates the environment is bad (the current neighbors provide a lot of bad services). The reasons that bad neighbors are chosen are two-fold: (1) the neighbors are turning worse, and (2) the received ratings are wrong so that the *SP* itself is misled in the neighbor selection. Increasing $S_N$ is very useful to solve the problem when the neighbors turn bad, because the larger the neighbor list is, the larger the probability that a good *SP* is included in the neighbor set. However, increasing the size of neighbor list incurs a considerably increase of the storage cost (each additional neighbor in the neighbor list will lead to the installment of one rating queue and one history table). Moreover, it will bring more network traffic when the number of objects (*SP*s) to be rated increases for each *SP*. We define a severe threshold $\theta_E$ (its value is 0.5 in our simulation) to measure the quality of the environment. When $\alpha > \theta_E$, the environment is thought to be severe so that the new size of the neighbor set $S_N^{new}$ will be enlarged to $(1+\alpha) * S_N^{old}$. When $\alpha = \theta_E$, which means the healthiness of the environment is moderate, the neighbor list keeps the same size as before. When the environment turns good, implied by $\alpha < \theta$, the neighbor list will be shrunk to $Min(S_N^{old}-1, S_{Init})$ to reduce the cost of storage and traffic, where $S_{Init}$ is the initial size of the neighbor list. For the problem (2), decreasing the weight of rating is useful to inhibit the negative effect of bad ratings. When $\alpha \geq \theta_E$, the weight *W* is set to a fixed low value $\rho$, which is the weight of the rating. Simulation results in [6] suggest that if $\rho$ is set to a value between 0.2 and 0.3, the negative effect of the rating can be greatly inhibited with the acceptable sacrifice of the efficiency. If $\alpha < \theta_E$, the environment is healthy, so *W* is adjusted according to the quantity of the interaction-derived information. In this case, $W$ is defined as the temporal injection degree, i.e., the ratio of the number of cooperations $h$ to the size of the history table $S_H$ in a specific time.

### C. A Simple Economic Model

As mentioned in Section II-B, with the help of *a*PET, each ISP builds its own personalized trust map for its neighbor ISPs. Based on this trust map, each ISP picks out the good-quality neighbor, which increases the possibility to construct a good route and increase the success rate of the routing. Once the system gets stable, the ISP's selection of the next hop for the routing, and the user's selection of the server ISPs based on the success rate of the ISP service also get stable.

To support the economic phenomenon in ISP peering, we build a simple economic model on top of *a*PET and introduce the concepts of "currency," "currency exchange" and "currency ratio." Each ISP issues its own currency based on its service capacity. The cooperation among ISPs is implemented through currency exchange and redemption. In TRECON, two techniques have been used to introduce the incentives. First, for the ISPs providing the good services, their currency will have high exchange ratio. We associate the value of exchange ratio (`R`) with the trustworthiness value (`T`) using a function $f$, (`R` = f(`T`), in our simulation, `R` = `T`), through which currencies from bad-quality ISPs are devalued since their trustworthiness values are decreased. Finally their currencies lose the purchase power and they will be eliminated from the network. The second technique is self-adjusting the price of the service according

to ISP's overload. In our simulation, when the service number in a certain span is more than a threshold, an ISP can raise its service price, because it is reasonable to think that ISPs providing good services attract more users, and the good services deserve higher price.

### D. Two Strategies

Defining a selection function $\hbar$ to properly select the next hop $s$ plays an important role in TRECON. Two selection strategies are studied in this paper. The first one is to select the next hop with the highest trustworthiness value. The second strategy, the shortest path, is set as the baseline strategy for the purpose of comparison.

**1. Maximum Trustworthiness Value (TRU)** The trustworthiness value derived from *a*PET is a numeric value representing the personal view on the neighbor's quality. Selecting the next hop with the maximum trustworthiness value is the most direct strategy for the next hop selection. We briefly call this strategy as *TRU*. Suppose the neighbor set is denoted as $N$, and the corresponding trustworthiness set is $T$; for neighbor $i \in N$, the trustworthiness value is denoted as $T_i$. The selection function for TRU is then formalized as $\hbar \equiv i, i \in N \wedge T_i = \max(T)$. To avoid routing loops, we develop an assisting navigation system, similar to GPS, which helps choose the right next step. The neighbor with highest $T$ value in the forwarding direction is selected with high priority. The details of the navigation system can be found in our technical report [8].

**2. Shortest Path (SPA)** Shortest path routing is a optimal routing approach in terms of distance. Some important Internet routing designs like BGP choose the shortest path based routing as the default routing strategy. We choose this approach as the baseline. This strategy, simply denoted as *SPA*, is defined as $\hbar \equiv i, i \in N \wedge i \in \ell_{v_s,v_d}$, where $\ell_{v_s,v_d}$ stands for the node set in the shortest path from sender $v_s$ to destination $v_d$.

Different ISPs can have different strategies. In our technical report [8], three more strategies are also studied. To express the routing preference, each ISP only needs to send the preferred parameters on reliability or delay together with its request. ISPs along the route will choose the next hop based on these parameters from the original sender. If the routing preference is originated from end users, our approach then can provide end users the ability to affect the selection of the sequence of Internet service providers a packet traverses, close to the goal of NIRA [17].

## III. Design and Metrics

Building a simulator of the trusted ISP peering is one part of our contributions in this paper. To our knowledge, most of state-of-the-art simulators for ISP peering focus on the routing [3], [14], and few work has been done on the economic effects on ISP peering. Our simulator distinguishes itself from other simulators in integrating both routing and economic performance evaluations. In this section, we describe the design of the simulator in detail. We also propose several important performance metrics against which we evaluate the performance of our model.

### A. Simulator Design

We build our simulation platform using NetLogo [16], a very popular multi-agent simulation tool in the AI community which can easily model parallel and independent agents and simulate interactions among ISPs. We have developed a friendly GUI-based user interface to control the simulation, through which we can easily tune different parameters to set up different configurations.

ISP peering is a complicated inter-networking, and its performance can be affected by many factors. In the simulation, we try to simulate the real network from two angles:

**1. Links** Each ISP is connected with links. If the quality of the links is bad, the performance of ISP peering degrades. *Reliability*, *delay*, and *bandwidth* of the link are three parameters to evaluate the quality of the link. Considering the fact that most ISPs provide high bandwidth links, we assign a high bandwidth to each link in the simulation. Regarding the delay of the link, in the simulation we purely make it proportion to its physical length. So finally reliability ($\Re_l$) becomes the major parameter for the link quality. In the simulation, links are assigned a value in the range [0,1] as the link reliability. Each time the routing request passes through the link with probability $\Re_l$.

**2. ISPs** The quality of each ISP is the other important factor. An ISP (denoted as a node in the simulation) may be good, bad, or even malicious. Considering the real situation and the space constraints, we exclude the malicious case in the simulation. The *processing delay* and the *processing capacity* are two metrics used to evaluate the quality of one ISP. In real ISP peering, it is normal that one ISP equips with enough processing capacity. So in the simulation each ISP is assigned with a relatively large number for this factor so that it is not likely that an ISP gets jammed due to small processing capacity. The quality of each ISP is mainly reflected by the processing delay. More specifically, some ISPs are assigned a delay factor $\delta$ which is associated with the delay (length) of its ingress link, that is, if the delay of the ingress link is $D$, the processing delay of ISP is then $D * \delta$.

From these two angles, we generate the topology of ISP peering with different ISPs and links. Topology generation is a very big part of the simulation code. First topology is generated according to the global minimum and maximum degrees of an ISP (they can be adjusted in the GUI), and the maximum number of ISPs in each specified area. Then, we randomly select some ISPs and links to assign different delay and reliability values.

### B. Performance Metrics

To better evaluate the simulation results, we propose six evaluation metrics:

**1. Delay Index** $\wp_n$: Path delay is the sum of the delay along the path from the requester to the destination, which is calculated as $\wp = \sum(D_i * (1 + \delta_i))$. It is normalized as $\wp_n = \wp_s/\wp$ when T is calculated, where $\wp_s$ is the delay of the shortest path. If the routing fails in the middle, then $\wp_n = 0$.

**2. Path Reliability** $\Re$: Path reliability is the product of the

reliability of each link along the path, which is calculated as $\Re = \prod \Re_i$, where $\Re_i$ is the $i^{th}$ link along the path. If the package delivery fails, then $\Re = 0$.

**3. Cost for Forwarding Services** $\varsigma$: The cost for forwarding services is the sum of the cost for asking help from other ISPs, which is calculated as $\varsigma = \sum \varsigma_i$, where $\varsigma_i$ is the cost of the $i^{th}$ forwarding request.

**4. Earn from Forwarding Services** $\varepsilon$: Earn from forwarding services is the sum of profits for helping other ISPs to forward the package, which is calculated as $\varepsilon = \sum \varepsilon_i$, where $\varepsilon_i$ is the profit of $i^{th}$ forwarding service.

**5. Total Requests from User** $i$ $\lambda_i$: $\lambda_i$ is the total number of requests from users for ISP $i$.

**6. Net Profits** $\phi$: $\phi$ is the net profit after considering the profits from users, the profits from forwarding services, and the cost for forwarding services. $\phi$ is calculated as $\phi = P * \lambda + \varepsilon - \varsigma$. $P$ is the unit price for the client request. It is the major profit for the ISP. It is reasonable to make the assumption that $P$ is larger than the maximum price (cost) of the forwarding services. That is $\forall i, P > max(\varsigma_i, \varepsilon_i)$.

Among these metrics, delay index ($\wp_n$) and path reliability ($\Re$) are the two direct metrics of paramount importance to evaluate the quality of route; Profits ($\phi$) are the metric to evaluate the economic performance of the TRECON framework.

### C. Apply aPET to ISP Peering

When *a*PET is applied to ISP peering, the ISPs now take the role of the *SP*. Since the neighborhood in ISP peering is quite stable, the neighbor list is relatively fixed. To give enough flexible support for the routing policy, each element of the rating queue and history table is no longer a single value, but a value pair: (Delay Index $\wp_N$, Path Reliability $\Re$). To calculate the trustworthiness value, we change the rating value $R$ in the formula $T = W * R + (1 - W) * I$ in Figure 1 to $R = (\omega_d * (\Sigma \wp_{n_R}) + (1 - \omega_d) * (\Sigma \Re_R))/S_R$, and $I$ changes to $I = (\omega_d * (\Sigma \wp_{n_I}) + (1 - \omega_d) * (\Sigma \Re_I))/S_H$, where $\omega_d$ is the routing preference weight for the delay and correspondingly $1 - \omega_d$ for the reliability. One complete routing event will incur all the ISPs in the routing path to update the information in the history table and the rating queue. Suppose the routing path is $a \rightarrow b \rightarrow c \rightarrow d$. If finally the routing succeeds, $a$ will update the related information of $b$, $b$ will update the related information of $c$, and so on. When the routing fails in the link between $b$ and $c$, only $a$ needs to update $b$'s related information. The sender $a$'s preference, for example $\omega_d = 0.2$, is passed along the routing path when the routing starts, and the intermediate ISPs in the routing path derive the trustworthiness according to sender's routing preference (not their own preferences), and select the best next hop under *TRU* strategy.

## IV. Experimental Analysis

We aim to develop an efficient framework for the ISP peering which combines the routing structure, the adaptive trust model, and the trust-based economic model. To show the efficiency of TRECON, we use simulation to compare the performance of the *TRU* strategy to *SPA* strategy from the perspectives of delay, reliability, and economic effects. In this section, we present the detailed analysis. Our simulation is a round-based simulation. In each round (step) of the simulation, a certain amount of the service requests are generated. This number series follows a Poison distribution. The details of the simulation parameters are listed in Figure 2.

| **TRECON related parameters** | | |
|---|---|---|
| **Settings** | | **Illustrations** |
| $\omega_d$ | 0.2 | Weight of delay |
| $\omega_p$ | 0.2 | Weight of price |
| $S_E$ | 10 | Size of the environment alert queue |
| $S_H$ | 6 | Size of the history list |
| $\rho$ | 0.5 | Alert threshold |
| $S_N$ | Fixed | Number of neighbors |
| $S_R$ | $S_N$ | Size of the rating queue |
| **Topology related parameters** | | |
| **Settings** | | **Illustrations** |
| $N_I$ | 200 | Number of ISP |
| $N_L$ | 271 | Number of links |
| $U_h$ | 18 | Number of highly unreliable links |
| $U_m$ | 10 | Number of medium unreliable links |
| $U_l$ | 14 | Number of low unreliable links |
| $D_h$ | 13 | Number of highly delayed ISPs |
| $D_m$ | 18 | Number of medium delayed ISPs |
| $D_l$ | 3 | Number of low delayed ISPs |
| $C$ | 200 | Processing capacity of ISPs |
| $B$ | 200 | Bandwidth of links |
| **Other parameters** | | |
| **Settings** | | **Illustrations** |
| $\mu$ | 100 | Mean of Poison distribution |
| $N_S$ | 1000 | Total rounds in one simulation |

Fig. 2. Simulation settings and their illustrations.

### A. Path Delay and Path Reliability

Delay is an important metric to evaluate the quality of routing. *SPA* adopts the global optimal shortest path design strategy, and should be better than *TRU* in terms of delay. However, the delay can be incurred not only by the length and bandwidth of the link, but also by the traffic jam, the process delay of the node, the reliability of the link, and so on. Figure 3 (a) shows the delay statistics from the angle of

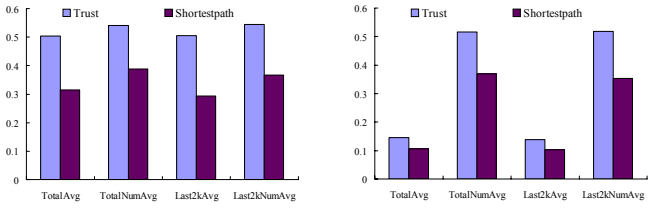



Fig. 3. Comparison of the delay index and path reliability: (a) The bar plot of the average delay index. (b) The bar plot of the average path reliability.

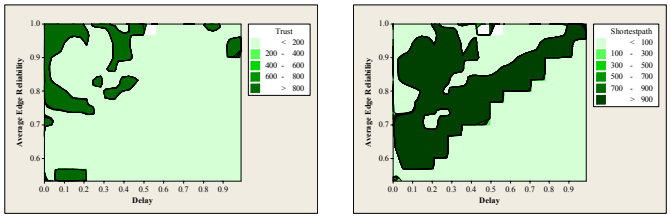
the whole system, that is, the average delay index of all user requests. In Figure 3 (a), there are totally four columns, and each column contains two bars which are the average delay index corresponding to two strategies. "TotalAvg" is the value derived from the whole process of system running, which is calculated as $\frac{\sum(\wp_{nj})}{\sum(\lambda_i)}$. $\sum(\wp_{nj})$ is the sum of delay index for all requests. "TotalNumAvg" is the success rate of all requests, which is calculated as $\frac{\sum(\lambda_{si})}{\sum(\lambda_i)}$. $\sum(\lambda_{si})$ is the total number of successful routings. "Last2kAvg" and "Last2kNumAvg" have the same meaning as "TotalAvg" and "TotalNumAvg",

but they only concern with the last 2000 client requests. For *TRU* strategy, the system may need time to get stable, so "Last2kAvg" and "Last2kNumAvg" reflect the performance when the system gets stable. Figure 3 (a) tells the fact that *TRU* has lower delay (higher delay index) than the optimal *SPA* all around. For the first column "TotalAvg", the average delay index reaches to 0.503, which beats *SPA* with value 0.314 by 60.2%. The value (overall success rate) difference in "TotalNumAvg" is smaller, but *TRU* is still can beat *SPA* by 39.3%. These two numbers increase to 72.4% and 48.2% respectively corresponding to the columns of "Last2kAvg" and "Last2kNumAvg". Since "Last2kAvg" and "Last2kNumAvg" are the data in the stable stage, it is more convincing that when the system gets stable, *TRU* is even better in improving the routing delay when the delay is not just decided by the path distance. Observing carefully about the average delay index, we find that there is almost no performance difference for *TRU* between the whole system running and the last 2000 client requests. This shows that, *TRU* converges very fast from the angle of the delay, which is preferred in the routing design.

Figure 3 (b) compares the path reliability between *TRU* and *SPA*. The meanings of the x-axis (columns) are similar as Figure 3 (a), while the y-axis is about path reliability instead of delay index. In the real network, a path may be the shortest in terms of geography but with low reliability. Basically, From Figure 3 (b) we can observe similar results of as those of delay: *TRU* is still obviously better than *SPA*; in the last 2000 requests, *TRU* has better results, while *SPA* degrades a little. However, the result difference between *TRU* and *SPA* is not that big as in the case of the delay analysis. The reason is the calculation of the final reliability value is the product of reliability of all links in the routing path. The "product" operation will make the value of reliability very small when there is a link with low reliability, so its value difference between *TRU* and *SPA* also decreases relatively. Normally the longer the path is, the more possibility to have an unreliable link in the path. Since *SPA* is taking the global shortest path, it is expects to have less links in the routing path for each service request, and thus should have less chance to get unreliable links into the production. But we still find that *TRU* is much better than *SPA*. This is because the reliability is part of the factor to derive the trustworthiness value. *TRU* can avoid the path with zero or very low reliability, which is out of the capacity of *SPA*. Hence selecting the neighbor with higher trustworthiness value also makes the path selection more reliable.

### B. Effects on User Decision

Users normally choose and change ISPs based on their service quality. In order to simulate this adaptive ISP selection of users, the user requests are assigned to the ISPs according to the success rate from high to low in the simulation. Only about 50% ISPs can get the service requests in each round. Users prefer the strategy which can help to send the requests to the ISPs with high quality. In our simulation, the ISP with high quality embodies in two aspects: the ISP's own quality (i.e., low processing delay) and the quality of the ISP's environment (i.e., the link quality of neighbor links). We use contour plots to show the relationship between the number of user requests and the qualities of ISPs that include two parts: *the delay of ISPs* and *the average link reliability*, as illustrated in Figure 4. The colors in the figure represents the quantity of requests. From Figure 4, we do see that, with *TRU*, most of the user requests go to the ISPs with high average link reliability (ALR) and low processing delay. The requests mainly reside in the area with delay $< 0.5$ and ALR $> 0.7$, i.e., the left upper corner. In the case of *SPA*, the user requests are distributed more evenly; considerable requests are sent to the ISPs with high processing delay or low ALR. From this, we conclude that, the strategy *TRU* is much better than *SPA* to direct the user requests to the ISPs with high quality. Another fact we can get from Figure 4 is, the highest level of request number (darkest part in the B/W mode) in *TRU* is just 800, while this value in *SPA* is 900. It shows that *TRU* has good property of load balance to distribute the user request more evenly.

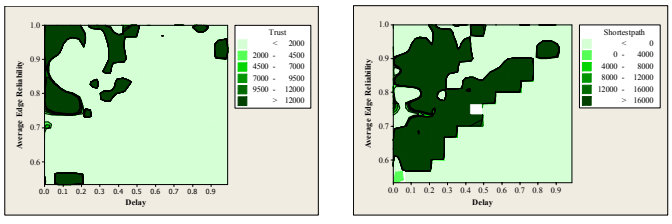



(a) (b)

Fig. 4. Contour plot of the total number of client requests vs. the average edge reliability and node delay:(a) *TRU* and (b) *SPA*.

### C. Economic Effects Analysis

Serving user requests are the source of economic profits for ISPs. Attracting more user requests means more profits. One of the potential advantages for introducing the economic model in TRECON is making the good ISPs receive more requests from users, and thus gain more profits than those low-quality ISPs. The two subfigures in Figure 5 are the contour plot of net profits *vs.* the ISP delay and average link reliability. It can be seen that in Figure 5 (a), most of the ISPs with large amount of profits are the ISPs with high quality (ALR $> 0.74$ and processing delay $< 0.5$). However under SPA (Figure 5(b)), some ISPs with low ALR or high delay still get a lot of profits. Thus we conclude that *TRU* is much better than *SPA* to stimulate ISPs to provide good-quality services.




(a) (b)

Fig. 5. Economic effect analysis: (a) *TRU* and (b) *SPA*.

## V. Related Work

This work builds upon many previous efforts in two categories: (a) Internet routing and (b) trust and economic models.

**Internet Routing:** BGP [12] has the problem of how to build an efficient routing structure and support flexible routing policies. Matthew and Jennifer [2] observe that most of the mystery in BGP comes not only from the protocol complexity but also from a lack of understanding of underlying policies and the problems that ISPs face. Feigenbaum, Sami and Shenker [4] theoretically study the policy-routing problem. They find that computing an optimal set of BGP-based routes is NP-hard, and then present a strategyproof, polynomial-time computable mechanism for welfare-maximizing routing over the restricted domain. TRECON have no assumptions on the restricted domain, and approximately optimize the routing for ISPs with different routing preferences.

HLP [14] aims to more scalability, better isolation, and faster convergence than the current BGP routing for the next generation inter-domain routing. TRECON and HLP share the same goal of attacking the problem of scalability, isolation, and convergence of routing, and they also share several common design philosophies: changing the granularity of routes from prefix based to AS based; modifying the flat routing structure; and using hybrid routing instead of path vector based routing. However, in [14], there is no detailed approach to specify how to build their hierarchical routing structure, while in TRECON, a detailed clustering algorithm has been proposed to build the cluster routing structure. On the other hand TRECON focuses on the concrete policy design based on the trustworthiness and economics, HLP focuses more on the adaptability to different policies. Nexit [9], a negotiation framework to support negotiation-based routing between neighboring ISPs, is similar to our work. Both of them are to build the cooperation routing between competing entities. However Nexit relies on the negotiation to reach the agreement on the path selection; in TRECON, each ISP makes the decision independently base on the available trustworthiness and price information.

**Trust and Economic Models:** Our work builds on a lot of previous work on the trust and economics in P2P network. We envision that trustworthiness should be fundamental to the design of any economic based P2P resource sharing. Numerous economic models including microeconomics and macronomics principles for resource management have been proposed in the literature [1], [5], [15], and various criteria are used for judging the effectiveness of an economic model, including social welfare, stability, and computation efficiency. However, for these work targeting for P2P environments, the trustworthiness of peers is either neglected or treated as an optional factor. The previous work also separates computational economy from the trustworthiness of participating peers, which is a necessary component to make the economic model feasible and reliable in an open environment. In TRECON, the adaptation of currency ratio between two peers will be determined by the trustworthiness values.

## VI. Conclusions

In this paper, we propose a trust-based economic framework TRECON to attack the open problems in ISP peering for NGI. The analysis shows that the proposed trust-based routing approach has significant advantage over the traditional shortest path approach in terms of reducing the delay, increasing the reliability, balancing the load among ISPs and links, directing user requests to ISPs with good quality and maximizing the profit of good ISPs. We envision that our TRECON can be incrementally deployed on the Internet and works well other solutions for NGI. Our next step will focus on the improvement of the economic model and pricing model, and building a prototype on PlanetLab [11].

## References


[1] Y. Amir, B. Awerbuch, and R. Borgstrom. A cost-benefit framework for online management of a metacomputing systems. *Proc. of the first Inernational Conference on Information and Computational Economy*, Oct. 1998.

[2] M. Caesar and J. Rexford. Bgp policies in isp networks. *to appear in IEEE Network Magazine, special issue on interdomain routing*, Nov. 2005.

[3] N. Feamster, J. Winick, and J. Rexford. A model of bgp routing for network engineering. *Proc. ACM SIGMETRICS*, Jun. 2004.

[4] J. Feigenbaum, R. Sami, and S. Shenker. Mechanism design for policy routing. *Proc. of ACM Symposium on Principles of Distributed Computing (PODC)*, 2004.

[5] A. Lazar and N. Semret. Auctions for network resource sharing. Tech. Rep. TR 467-97-02, Computer Science Department, Columbia University, Feb. 1997.

[6] Z. Liang and W. Shi. Enforcing cooperative resource sharing in untrusted peer-to-peer environment. *ACM Journal of Mobile Networks and Applications (MONET) special issue on Non-cooperative Wireless networking and computing* 10(6):771–783, Dec. 2005.

[7] Z. Liang and W. Shi. Performance evaluation of different recommendations aggregation schemes in reputation systems. *in Proc. of first IEEE International Conference on Collaborative Computing: Networking, Applications, and Worksharing (CollaborateCom '05)*, Dec. 2005.

[8] Z. Liang and W. Shi. Trecon: A framework enforcing trusted isp peering. Tech. Rep. MIST-TR-2006-006, Department of Computer Science, Wayne State University, Mar. 2006.

[9] R. Mahajan, D. Wetherall, and T. Anderson. Negotiation-based routing between neighboring ISPs. *Proc. of Networked Systems Design and Implementation (NSDI)*, May 2005.

[10] Nsf report of workshop on overcoming barriers to disruptive innovation in networking, Jan. 2005, `http://planet-lab.org/doc/barriers.pdf`.

[11] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the internet. *Proc. of ACM First Workshop on Hot Topics in Networks (HotNets-I)*, Oct. 2002, `http://www.planet-lab.org/pubs/hotnets.pdf`.

[12] Y. Rekhter and T. Li. A border gateway protocol 4 (bgp-4), Mar. 1995, `http://www.faqs.org/rfcs/rfc1771.html`.

[13] J. Shneidman, C. Ng, D. C. Parkes, A. AuYoung, A. C. Snoeren, A. Vahdat, and B. N. Chun. Why markets could (but don't currently) solve resource allocation problems in systems. *Proc. of the 10th USENIX Workshop on Hot Topics in Operating Systems (HotOS-X)*, June 2005.

[14] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. HLP: A next-generation interdomain routing protocol. *Proc. of ACM SIGCOMM*, Aug. 2005.

[15] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta. Spawn: A distributed computation economy. *IEEE Transactions on Software Engineering* 18(2):103–117, 1992.

[16] U. Wilensky. Netlogo, 1999, `http://ccl.northwestern.edu/netlogo`.

[17] X. Yang. Nira: A new internet routing architecture. *Proc. of ACM SIGCOMM FDNA 2003 Workshop*, Aug. 2003.