TRECON: A Trust-Based Economic Framework for Efficient Internet Routing

Zhengqiang Liang and Weisong Shi, *Senior Member, IEEE*

***Abstract*—The fragility and the poor resilience of the Internet are manifested by the severe impact of network activities and the slow recovery after an earthquake damaged undersea cables and disrupted telephone and Internet access in East Asia in December 2006. Except the inefficiency of routing protocols, lack of efficient network monitoring mechanisms and lack of economic incentives to encourage service providers (SPs) to act cooperatively and promptly are other important reasons. In this paper, we build a trust-based economic framework called TRECON to address these open problems in Internet routing. The novelty of TRECON is combining an adaptive personalized trust model with an economic approach to provide independent trust-based routing among SPs. TRECON provides flexible policy support based on the trust-based economic mechanism so that autonomous organizations with varied interests and optimization criteria can be smoothly integrated together to achieve better adaptiveness and self-management. Through introducing the economic model, TRECON explores a new way to solve the economic problems and incentives issues in the collaboration among SPs. To show the flexibility of routing policies support, we propose four typical routing policies under the TRECON framework. We evaluate our approach by comparing these four trust-derived routing policies with the classical global shortest path routing approach. We find that the policy based on trustworthiness performs much better than all other policies under different network topologies in terms of delay, success delivery rate, and economic effects.**

***Index Terms*—Adaptive PErsonalized Trust (aPET) model, adaptive trust model, economic model, incentives, Internet routing, Internet service provider (ISP).**

I. Introduction

A PAIR of powerful earthquakes off the coast of Taiwan damaged undersea cables and disrupted telephone and Internet access in Asia on December 26, 2006. Although service providers (SPs) tried every means to restore Internet access by rerouting traffic, there were still 97.58% Internet users who reported that their Internet access was disrupted, according to a survey from Sina.com on the next day after the quake [1]. This incident highlights the fragility of the Internet, and the poor resilience of Internet routing (including intradomain routing and interdomain routing), which are caused by two main reasons: 1) the Internet routing protocol is not flexible and fast enough to self-adjust the routing when the network topology is changed and 2) the SPs from the routing substrate are not able to act promptly and cooperatively because of lack of efficient network monitoring mechanisms and their economics conflicts.

To bring the Internet back to where it is expected to be (i.e., the Internet was designed in part to provide a communications network that would work even if some of the sites were destroyed by nuclear attack [2]), redesigns on routing are needed. Correspondingly, the new design should endow the Internet with more flexibility and self-adaptiveness in routing in the face of topology change severely, and it can provide a fair and reciprocal collaboration among SPs. The report on a recent National Science Foundation workshop [3] reveals that the future design for the Internet must take competition and economic incentives into account. It is worth noting that the notion of SP in this paper is more general than the Internet SP (ISP). An SP could be an ISP, an autonomous system, a peer in overlay networks, or even a router, depending on the specific application scenario of TRECON. We use this notion to generally represent an independent routing unit under the TRECON framework.

We summarize that there are three requirements to build effective Internet routing.

1) *Distributed Network Monitoring:* Distributed network monitoring can catch the dynamics of the routing and the changes of the quality of SPs, so that each SP can make intelligent routing decisions to improve performance, self-organization, and routing reliability.
2) *Flexible Policy Expression:* The flexible policy expression is helpful and necessary to let the new routing architecture be accepted. The SP should be allowed to express their decisions to manage the routings, and clients should be allowed to express their demands for the routings.
3) *Healthy Incentives-Based Environment:* In Internet routing, every SP aims to get economic benefits by attracting clients as much as possible. Clients need good services from good SPs. A healthy environment of Internet routing should direct requests of clients to the good SPs to maximize the satisfaction of both SPs and clients.

In this paper, we propose TRECON, a TRust-based ECONomic framework which meets the aforementioned requirements to enforce the next-generation trust-based Internet routing. TRECON consists of two major components: *an a*daptive PErsonalized Trust (*a*PET) *model and a trust-based economic model*, which can be viewed as two layers in TRECON, as shown in Fig. 1. TRECON makes use of the trustworthiness information provided by a novel *a*PET model to evaluate the quality of SP. For each SP, *a*PET flexibly combines self-experiences and other SPs' feedbacks to derive the trustworthiness of neighbors. With the help of the trust inference, the quality of SPs can be quantified, which makes the routing selection easy to implement. Since the trustworthiness is related to

Manuscript received October 13, 2007; revised March 1, 2009. First published November 3, 2009; current version published December 16, 2009. This paper was recommended by Associate Editor K. W. Hipel.

The authors are with Wayne State University, Detroit, MI 48202 USA (e-mail: sean@wayne.edu; weisong@wayne.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2009.2030730

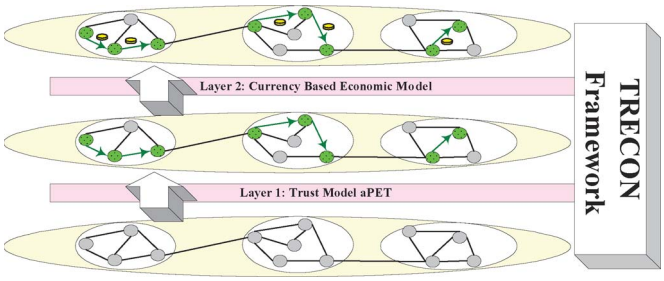


Fig. 1. Two layers of the TRECON framework. The bottom layer is the trust model *a*PET, and the top layer is the currency-based economic model.

service quality, TRECON is promising for *QoS support*, which is one of the preferable requirements of the future Internet. TRECON employs an economic model to meet the economic requirements in the Internet routing. Each SP actually is an independent economic entity. They mutually collaborate but compete at the same time. Employment of economic model in the routing can effectively coordinate SPs. Ignoring economic consideration is one of the main drawbacks of current Internet protocol stack. SPs can set up their own routing policy based on the trustworthiness and economic information provided by TRECON. Having this flexible policy support, we set up a design point between the hot potato routing and the cold potato routing [4]. The autonomous organizations with varied interests and optimization criteria are then smoothly integrated together to achieve better scalability, isolation, and self-management. By introducing the trust and economic model, TRECON aims to *fit the design of the future Internet*.

To comprehensively evaluate our trust-based routing (*TRU*) policy, we propose other three trust-related routing policies together with the optimal global shortest routing (*SPA*) policy in the same network topology. We find that the *TRU* policy has best performance among all policies. The results indicate that, using trustworthiness information to direct routing is promising in the design of the next-generation Internet.

II. *a*PET Model

In TRECON, we argue that the routing should be directed by the trustworthiness information, which is provided by a novel *a*PET model. Intuitively, having a trustworthiness map (also known as reputation) of participating SPs is very helpful for collaboration. However, it is difficult to calculate the trustworthiness value due to the independence and dynamic behaviors of SPs and the absence of an effective security mechanisms. Unlike the *Central* [6] and *Transitive* [7] models, we argue that it is important to build an *Independent* model for Internet routing. In the independent model, it is not necessary to force all SPs to agree on one global trustworthiness value and the transitive relationship (e.g., if A likes B, and B likes C, then A will like C). Every SP has its own view on trustworthiness values of others and would not base on the view of others directly. This model matches the autonomy and self-management of the open distributed environment extremely well. *a*PET is such an independent model.

In *a*PET, similar as [8], the trust is defined as *the subjective expectation an individual A has about another individual B to perform a given action as good as expected in a certain time*. *a*PET is built based on our previous PET model [9], a personalize trust model proposed in the context of peer-to-peer (P2P) systems, and our thorough analysis to the main rating models in current researches [10]. Therefore, *a*PET is designed based on many data analysis and experience from our previous works. Different from other trust model including PET, *a*PET can self-adaptively change the weight for trustworthiness derivation according to the change of the environment. In the Internet routing, the trustworthiness information of neighbors provided by *a*PET is used to help find a good-quality route. We include the key observations from our previous works [9], [10] as follows.

Observation 1: Rating is not always as helpful as what we expect, particularly when the system is facing bad raters (we call the SP sending out the rating as a rater) and SPs with highly dynamic behaviors.

Observation 2: When the target environment has many dynamics and malicious peers, employing the simple average rating aggregation algorithms is better than the complex rating algorithms considering the overall effects of both the performance and implementation costs.

Observation 3: When the environment is getting worse (many bad raters, bad and dynamic SPs), lowering the weight of ratings and increasing the size of the neighbor table are very helpful to improve the performance of the trust model. These observations are the main design instructions for *a*PET.

Moreover, 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, particularly under the severe environment.

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

A. Model Design

In *a*PET, the trustworthiness `T` is derived from two parts: interaction-derived (also called self-experience) information `I` and rating `R`. Interaction-derived information is achieved through direct experience with other SPs, which is regarded as a kind of reliable sources for the derivation of the trustworthiness value. However, this kind of information cannot bring the efficiency to the trust model. The rating from other entities, which can help to discover the quality of other SPs even without direct transactions, is introduced for the efficiency purposes. However, rating is not reliable due to the dynamics of environment and the malicious raters. Integrating these two kinds of information enables *a*PET to inherit its advantages while inhibiting its disadvantages.

Fig. 2 shows an overview of the *a*PET. To be worth noting, although *a*PET is introduced for the Internet routing in this paper, it is a general trust model which can be applied to other P2P-like applications like Web server peering and P2P file sharing. In Fig. 2, every SP has its own neighbor set, as shown in the left side. The neighbor selected for the collaboration is called the *collaborator*. 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 Fig. 2, there are three neighbors: SP A, B, and C. Correspondingly, there are three elements in the neighbor list,

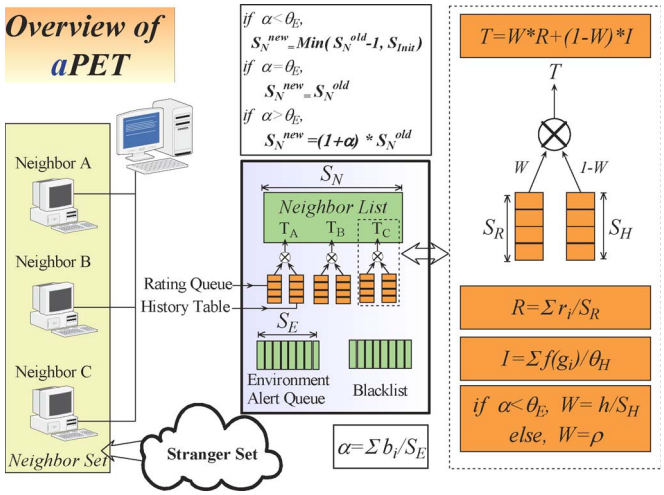


Fig. 2. Overview of *a*PET model.

each of which includes the fields of SP ID, the trustworthiness value, and the ripple level number (the ripple level number is explained in Section II-B). For every neighbor, two local data structures, namely, rating queue and history table, are used to store the rating and interact-derived information, respectively. To meet the demand of *Re.1* which requires the trust model to be adaptive as the change of the environment, the other global data structure, environment alert queue, is employed to sense the quality of the environment. Neighbor list, rating queue, history table, and environment alert queue are all first in first out queues. Their sizes are denoted as S_N, S_R, S_H, and S_E, respectively. As described in *Observation 2*, it is not worth paying so much energy in the rating aggregation algorithm design considering overall effects of both the performance and implementation cost [10]. In *a*PET, the simple average scheme is used to aggregate the ratings. The rating r_i is the ith 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. Since `I` stands for the reliable information, it deserves more weight when the environment is turning worse.

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 (based on *Observation 3*). We introduce the environment-aware factor α to guide the adaptiveness. The environment alert queue is used to sense the surrounding environment changes. It records the quality of the most recent services from collaborators. The ith element b_i in this queue can be either 0 (good service) or 1 (bad service). α is defined as the proportion of bad services in the most recent interval, i.e., $\sum b_i/S_E$. A large α indicates that the environment is bad (the current neighbors provide many bad services). There are two reasons why bad neighbors are selected: 1) the neighbors are turning worse and 2) the received ratings are wrong so that the SP is misled during the neighbor selection. Increasing S_N is helpful to solve the problem when the neighbors turn bad, because the larger the neighbor list is, the higher probability to have a good SP in the neighbor set will be. However, increasing the size of neighbor list can incur significant storage cost (each additional element in the neighbor list will lead to the installment of one rating queue and one history table). Moreover, it can bring more network traffics when the number of objects to be rated increase. We define a severe threshold θ. When $\alpha > \theta$, the environment is thought to be severe so that the new size of the neighbor set S_N^{new} will be enlarged to $(1+\alpha) * S_N^{\text{old}}$. When $\alpha = \theta$, which means that 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 $Min(S_N^{\text{old}} - 1, S_{\text{Init}})$ 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 $\alpha \geq \theta$, the weight W is set to a fixed low value ρ, which is the weight of the rating. Simulation results in [10], [11] suggest that if ρ is set to a value between 0.2 and 0.3, the negative effect of the rating can be greatly inhibited with the acceptable degree of efficiency sacrifice. If $\alpha < \theta$, the environment is healthy, and then 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 collaborations h to the size of the history table S_H in a specific time.

III. TRECON Framework and Policies

The TRECON framework combines trust inference and a market-based approach to introduce efficiency, incentives, and profits to the Internet routing. *a*PET is the underlying trust inference infrastructure of the TRECON framework, on top of which an economic component is introduced. In addition, to apply TRECON framework in ISP peering, we introduce a cluster approach to make the whole ISP network can be travel sequentially in the high level.

A. Simple Economic Model

As mentioned in Section II, with the help of *a*PET, each SP builds its own personalized trust map for its neighbor SPs. Based on the trust map, each SP can pick up the good-quality neighbor as the next hop. It increases the possibility to construct a good route and improves the success rate of routing. To support the economic phenomenon in the Internet routing, we build a simple economic model based on *a*PET. The economic model decides the payment (P_a) when one SP asks helps from its neighbors to forward the packets. There are two principles to fix the payment of a forwarding service. First, the neighbors, who have large traffic volume, may raise the price (P_r) of the service, because the large traffic volume may imply that the forwarders are providing good quality services; second, the requester with a higher trustworthiness value should pay less than those with low trustworthiness values, because the former must provide good forwarding services to the forwarder before so that it has higher trustworthiness values in the eyes of the forwarder. These two principles are mathematically described in

$$P_r = \min\left\{\hat{P}_r, \frac{C_t}{C_l}\right\} \tag{1}$$

$$P_a = \min\left\{\hat{P}_a, \frac{P_r}{\bar{\mathrm{T}}}\right\} \tag{2}$$

where $\hat{P}_r$ and $\hat{P}_a$ are the upper bound of price and payment, respectively; C_t and C_l stand for the total capacity and the remaining capacity of a next-hop SP; and $\bar{\mathrm{T}}$ is the trustworthiness

value of the requester from the viewpoint of the next-hop SP. Using these two equations, we define the payment as a function of the environmental dynamics (i.e., processing capacity of the next-hop SP) and the historical behaviors (i.e., the trustworthiness value of the requester in the eye of the next-hop SP). Therefore, the payment is totally decided by next-hop SPs. With our economic model, two economic characteristics are introduced.

1) *Reciprocal collaboration:* When one SP attracts too much traffic, it may enforce the future traffic to transfer to other SPs by increasing its price of the forwarding service, if the payment is considered in the next-hop SP selection. Through this way, we can distribute the benefits from the forwarding service to others while maintaining the traffic balance of the system and equipping the system with capability of congestion control. Although in a short period, the SP providing good services may lose some benefits by this traffic transfer, but in the long run, the whole system can get benefits by this kind of reciprocal collaboration.
2) *Incentives:* The experiment results in Section V show that the whole system can get benefits from the angles of reducing the routing delay, increasing the routing success rate, and balancing the load among SPs and links. This can encourage all SPs to collaborate under TRECON although they compete at the same time. For a forwarder, a requester with low trustworthiness value needs to pay more than other requesters who have higher trustworthiness value; end users may switch from the SPs with low trustworthiness value (which implies low service quality) to those with high trustworthiness value. It makes the SPs providing low-quality services suffer from the economy loss. Therefore, all SPs have incentives to improve their service quality to gain more economic profits.

Note that in this paper, we only want to *preliminarily* show the effectiveness and correctness of the trust-based economic model in the Internet routing. Other issues such as designing a comprehensive pricing and payment model will be discussed in the paper related to our *M-CUBE* [11] model (Multiple CUrrency-Based Economic model) instead of this paper. *M-CUBE* is a more complicated and comprehensive economic model which is under parallel development currently. In this model, we introduce the concepts of "currency," "currency exchange," and "currency ratio." Each SP issues its own currency based on its service capacity. We are expected to finally integrate this currency model into the TRECON framework to replace the simple economic model we previously mentioned.

B. Next-Hop Selection Policies

Routing problem is a coordination problem. The routing decision can be single-hop based or multihop based. Considering the scalability, computation overhead, and adaptability, single-hop-based routing scheme is more reasonable, which is used in this paper. In this scheme, it is important to define a selection function $(\hbar)$ to choose next hop (s). With the support of trust and economic information, TRECON provides many *flexibility* to support different policies for the next-hop selection. We are advocating the *TRU* policy. However, is the trust-directed routing good enough? In order to find out the best routing policy under our TRECON framework, and show our framework's flexibility, we propose five different routing policies totally including *TRU* and compare them comprehensively. To help to understand the results, the standard shortest path routing (*SPA*) policy acts as the baseline for the purpose of comparison. The reason to choose *SPA* as the baseline is as follows: It is the classic routing algorithm in the textbook and many routing protocols like open shortest path first (OSPF) are SPA style. The routing paths in *SPA* are calculated through exhausted computing offline based on the path length, and they are absolutely shortest. Five typical policies are described as follows.

1) *Maximum Trustworthiness Value (TRU):* The trustworthiness value derived from *a*PET is a numeric value representing the personal view on the service quality of neighbors. Selecting the next hop with the maximum trustworthiness value is the most direct policy for the next-hop selection. We briefly call this policy as *TRU*. Suppose the neighbor set is denoted as N, and its corresponding trustworthiness value set is T; for neighbor $i \in N$, its trustworthiness value is denoted as T_i. Then, the selection function for this policy is formalized as $\hbar \equiv i, i \in N \wedge T_i = \max(T)$.
2) *Minimum Ripple Level (RIP):* In some systems like P2P, dynamics is the most distinguishing characteristic. When the churn rate of the system is high, the trustworthiness value cannot reflect the real situation because there is a delay for the trustworthiness value update. We introduce a ripple model (*RIP*) to handling the situation with high churn rate. The neighbors providing continuous r bad services are assigned with ripple level r. When using $\boldsymbol{RIP}$, nodes with lowest ripple level and higher trustworthiness value if ripple level is the same will be selected in priority. Let R stand for the ripple level. Then, the neighbor set can be partitioned into $N = \bigcup N_{R=r}$, $r = 1, 2, 3, \ldots$, where $N_{R=r}$ denotes the neighbor set with the ripple level $R = r$. Correspondingly, the trustworthiness set is partitioned as $T = \bigcup T_{R=r}$, $r = 1, 2, 3, \ldots$. The selection function for this policy is formalized as $\hbar \equiv i$, $i \in N \wedge T_i = \max(T_{R=1})$, in which the neighbor with maximum trustworthiness value in the level 1 (inner-most level) is selected.
3) *Minimum Payment (MPA):* TRECON introduces the concepts of price and payment. Based on these two concepts, we can simulate another prevailing selection policy, *MPA*, to route with economic consideration. In *MPA*, the neighbor with the lowest payment rate P_a for forwarding service is selected as the next hop. Formally speaking, the selection function for this policy is $\hbar \equiv i$, $i \in N \wedge P_{ai} = \min(P_{an})$, where P_{an} is the payment set about the neighbors.
4) *Combination of Trustworthiness and Payment (COM):* *TRU* and *MPA* are two completely different selection policies. Combining both policies, we get another policy which is denoted as *COM*. In the *COM* policy, each neighbor i has a combination value C_i, which is calculated as

$$C_i = w_p * P_{ai}^{-1} + (1 - w_p) * T_i \tag{3}$$

where $0 < w_p < 1$ is the weight of payment. Note that in our simulation, the minimum payment is one. In order to normalize the combination value, we use P_{ai}^{-1} for

the combination instead of P_{ai}. Let $C = \{C_i | i \in N\}$, and then the selection function is defined as $\hbar \equiv i$, $i \in N \wedge C_i = \min(C)$, i.e., the neighbor with the highest combination value is selected.

5) *Shortest Path (SPA):* In the literature about routing algorithms, finding out the route with the shortest path is a very attractive goal. Some important Internet routing designs like OSPF choose the shortest path-based routing as the routing policy. Using this policy as the baseline to compare the performance is persuasive. This policy, simply denoted as *SPA*, is defined as $\hbar \equiv i$, $i \in N \wedge i \in \ell_{v_s,v_d}$, where ℓ_{v_s,v_d} stands for the node set in the shortest path from sender v_s to destination v_d.

Different SPs can have different policies. To express the routing preference in the four trust-related policies, each SP only needs to send the preference parameters like w_p and w_s together with its request. SPs in the middle of the route will choose the next hop based on these preference parameters from the original sender. Having this flexible policy support, we set up a design point between the hot potato routing and the cold potato routing [4] which allows users to enforce some controls in the routing procedure.

C. ISP Network Clustering

There are some topology restrictions when apply TRECON in the Internet routing to avoid routing loop and keep the routing table small. Path-vector routing is used in border gateway protocol (BGP) [12], which can avoid loops in routing. However, it also causes a severe problem that the routing table becomes oversized. Building a hybrid routing structure like hybrid link-state path-vector (HLP) [13] is promising to combine the advantages of both path-vector and link-based routing. Different with the hierarchical structure used in HLP, our basic idea to solve this emerging problem is to build the cluster routing structure to change the granularity of routes and extract the stable route within the network. Then, the routing is separated into two parts, *intercluster routing* and *intracluster routing*. The intercluster routing is path-vector based, and globally fixed after the clusters have been formed. All ISPs need to store this global intercluster routing table. Since the routing unit is based on clusters, the intercluster routing table can be much smaller than the routing table in BGP if the ISP number in the cluster is control to be larger than a large enough threshold. For the intracluster routing, each ISP locally decides the next hop within one cluster mainly based on the trustworthiness of the neighbors. Since the routing decision is not globally visible, the routing failure and update then can be limited to one cluster. Therefore, under the cluster structure, the routing has good isolation and fault tolerance properties in addition to avoid routing loop. Comparing to the traditional BGP, our cluster structure is more scalable because each ISP just needs to store a small part of information, including the small global intercluster routing table and the trustworthiness information about its neighbors (neighbors means the ISPs connected with physical connections.). There are multiple clustering choices can be used in TRECON. The basic requirement for clustering is to build an acyclic tree in the high level. In this paper, we will take the linear topology, the most simple acyclic tree, as example to introduce the concepts and algorithms. The clustering approach can be applied in any acyclic tree based on the linear clustering with some extensions, which will not be presented in this paper.

Some Notations: Before describing our algorithm, we need to explain several definitions. We assume that the whole graph is a connected undirected graph. If the graph is not connected, we can just apply the algorithm to each connected subgraph.

Definition 1: A *cluster* is a nonempty connected subgraph $(\mathrm{V}', \mathrm{E}')$ of graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$.

Definition 2: Suppose there is an algorithm $\mathtt{A}$ which can find out $\mathbf{k}$ clusters within graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$. $G_i = (V_i', E_i')$ is the ith cluster, where $i \leq \mathbf{k}$, $|V_i'| \geq 1$, and $|E_i'| \geq 0$. If $(\{G_i\}, E - \bigcup E_i')$ is a acyclic tree (a line in current approach), graph G is called clusterable, $(\{G_i\}, E - \bigcup E_i')$ is called one *cluster choice* of graph G, and $\mathtt{A}$ is called the clustering algorithm.

Definition 3: Let $(\{G_i\}, E - \bigcup E_i')$ be one cluster choice of graph G. Then, $(\{G_i\}, E - \bigcup E_i')$ is called the High Level Cluster Graph of G. We simply denote it as HLCG(G). $(E - \bigcup E_i')$ is called the *bridge set* and denoted as E_B. For the nodes connected by the edges in $(E - \bigcup E_i')$, we call each of them as the *entry* of the cluster, and they consist of the cluster entry set V_E.

The clusterable graph has several unique advantages in the routing design.

1) In *HLCG(G)* (linear), the number of next hops is unique and unidirectional.
2) If the topology of each cluster and *HLCG(G)* are saved, the whole graph (routing information) can be easily rebuilt after the network suffers the large scale attack.
3) One cluster can be reclustered when its size is larger than a threshold, so clusterable graph is scalable and extensible. The value of the threshold can be fixed according to the demanding of network design and the network scale, which regulates the size of *HLCG(G)* and the cluster, respectively.

1) Clustering Approach: For one graph G, there may be different cluster choices. Finding an optimal (each cluster has equal number of ISPs) clustering algorithm is difficult or sometimes impossible. However, the goal of cluster does not focus on the clustering optimization (will be explored in the future work), but find out an applicable scheme to build a small size *HLCG(G)*, and each of the cluster is not overlarge. Therefore, the approximation approach is acceptable. Our algorithm is such an algorithm which builds on top of the assumption that the graph is connected. For the clusterable and connected graph, we propose a clustering algorithm, as shown in Fig. 3, to find out the clusters. The algorithm proposed is applied to the case where *HLCG(G)* is a path. If *HLCG(G)* is a circle, the circle has to be broken first by removing any one of the bridges. The general idea for the *Cluster* function is to find the bridge set E_B and the entry set V_E first, then using the entry point as the segment point of the cluster to build the cluster, and finally the function returns the cluster set C and bridge set E_B. After we get the clusters, all nodes in the graph need to store *HLCG(G)* as the intercluster routing. Combining the intracluster routing information introduced in Section II, the routing can be directed correctly. The advantage of the clustering approach is to distribute the enormous global information over all nodes, so that each node just needs to carry a very small part of the total information (a small interrouting table and a small

Cluster Nodes in A Connected Graph

$V_E = E_B = \varnothing$;
for all edge e in E **do**
  G' = G - {e};
  if IsConnected (G') = *False* **then**
    put e into the bridge set E_B;
    put entries a_e, b_e, or both into V_E if they are not in V_E;
  end if
end for
$V'_E = V_E$;
for all node $i \in V'_E$ **do**
  $C_i = C'_i = \{i\}$;
  for all node $j \in C'_i$ **do**
    $N = \varnothing$;
    for all j's neighbors $\tilde{j}$ **do**
      if $E_{(\tilde{j},i)}$ is not a bridge **then**
        $N = N \bigcup \tilde{j}$;
      end if
    end for
    N = neighbor set with all which are not in C_i;
    $C_i = C_i \cup N$;
    $C'_i = C'_i \cup N$;
    if $\exists$ node k, where $k \in N \wedge k \in V'_E$ **then**
      V'_E = V'_E - {k};
    end if
    $C'_i = C'_i - \{j\}$;
  end for
end for
return $C = \bigcup C_i$ and E_B;

Fig. 3. *Cluster* function.

intrarouting table), and the global information can be precisely reconstructed. For the disconnected graph, we can apply our clustering algorithm to each connected component.

The clustering algorithm is used in the bootstrap stage of the cluster construction. Clusters form the backbone of network and tend not to change. The newly joining nodes are assigned to available clusters. If a cluster becomes oversized, and this cluster is clusterable, two policies could be used to adjust the backbone. If reducing the workload of update is preferred, the oversized cluster can still stay as a cluster in the backbone, but within which two embedded subclusters will be generated (the cluster level increases by one). In this case, only the nodes in the oversized clusters need to add the intersubcluster routing information, and this update is invisible to the nodes in other clusters. If the balance of the backbone structure is preferred, the level number of subcluster needs to be limited. In this case, the oversized cluster can be split into two new clusters which are connected by a bridge if this cluster is clusterable. Although all nodes need to update *HLCG(G)*, there are only two places to be updated: First, replacing the oversized cluster G_i with two new small clusters G_{i1} and G_{i2}, and second, adding a new bridge E_{12} (the edge connects G_{i1} and G_{i2}) to the bridge set E.

In the ISP peering, the topology of ISP network has to be discovered first to make use of our clustering algorithm. In the simulation, the ISP topology used is supposed to be the physical geographic location. A few Internet mapping projects have used such tools to incorporate some notions of the geographic location in their maps, such as the Mercator Project [14] and the Internet Mapping Project [15]. These tools can be used to discover the ISP network topology.

ISP clustering can be combined with the autonomy from the point of organizations, politics, and geography. For example, all ISPs in one country is put in one cluster. Therefore, our algorithm is compatible with most of current ISP deployment requirements and can be incrementally deployed right now. Since the clustering algorithm only needs to be executed at the bootstrap stage of the system, it is acceptable to assume that the algorithm runs in a central node; then the result is broadcast to the other nodes, which is similar to the mechanism used by domain name system.

2) Nonclusterable Network: Not all graphs are clusterable, particularly for the graph with high connection degree and no bridges. In this case, we can try to find out the substructures which are clusterable and apply the *Cluster* function in Fig. 3 to form the cluster for those substructures. These substructures are embedded in the graph and will be considered as a single node for the upper level. The upper level nodes only need to know which cluster for the destination node resides, but do not care about complicated connections or embedded structures of the destination cluster. It means that the connection in the cluster is transparent for the upper level node. Then, a global map is built and stored in each node. If the size of one cluster is large, we can also use the clustering algorithm to make the second level cluster if the oversized cluster is clusterable, until the cluster size is satisfied. Then, the hierarchical cluster structure forms. For the hierarchical cluster structure, there is not too much difference for the implementation but just changes the routing table to reflect the hierarchical structure. Hence, in this paper, we consider only the flat cluster architecture. Since the size of cluster is limited, each ISP just needs to record limited path vectors within one cluster. For the routing table of the intercluster routing, its size is expected to increase slowly because the number of clusters increases far less than the number of ISPs. It is worth noting the worst case where the whole network degrades to just one cluster. In that case, the path selection will just consider the trust and location information. However, we may try to carefully remove some nonbridge links within the whole ISP network so to make it clusterable. Removal of different links can lead to cluster topology with performance variation. How to select the links to be removed and find optimal structure is a challenging problem, which will be explored in the future work. Another choice for the unclusterable networks is resorting to mechanisms resembling those of BGP to accommodate them.

IV. Experimental Methodology

To show the efficiency of TRECON, we use simulation to compare the performance of four trust-related policies with the *SPA* in terms of delay, success delivery rate, and economic effects. Although *SPA* cannot be really implemented in the Internet, we still choose *SPA* as the baseline because the comparison with *SPA* is more persuasive and measurable. In the simulation, we calculate the real optimal shortest path based on the path length offline, so *SPA* is expected to have better performance than the real BGP. Building a general simulator to evaluate different policies of Internet routing is one of our contributions in this paper. To our best knowledge, most of state-of-the-art simulators for Internet routing focus on the routing [13], [16], [17], and few work has been done on the economic

effects in the Internet routing. Our simulator distinguishes itself from other simulators in considering both routing policies and economic effects. Since in our previous work [9], [10], we have extensively studied the effects of malicious peers, and in the real SP routing case, most of SPs are selfish but not malicious, we will not extend our discussion about malicious SPs in this paper.

A. Topology Generation

We build our simulation platform using NetLogo [5], a very popular multiagent simulation tool in the artificial intelligence community which can easily model parallel and independent agents, to simulate interactions among SPs. With NetLogo, 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.

Until now, Internet topology generation is still a hard and unsolved topic. There are some available research results from Internet mapping projects like [14] and [15]. We are not going to attack this open problem in this paper. Instead, we generate a clusterable topology with some manual link removal and build the cluster with approach described in Section III-C in the simulation. We simulate the real network mainly from two angles.

1) *Links:* Each SP is connected with links (or edges). If the qualities of the links are bad, the performance of Internet routing degrades. *Delay*, *reliability*, and *bandwidth* of the link are three parameters to evaluate the quality of the link. Regarding the delay of links, in the simulation, we make it proportion to its physical length. To simulate the link reliability ($\Re_l$), in the simulation, each link is assigned with a value in range [0, 1] as the link reliability. Each time a packet passes through the link with probability $\Re_l$. The link reliability can also represent the comprehensive effect of packet loss because of not only link quality, but also other factors like traffic jam due to the shortage of link bandwidth and processing capacity of SP routers. The negative effects of the shortage of link bandwidth can be the delay or fail of the packet delivery. In the simulator design, we do not specifically assign a bandwidth to each link, but spread its negative effects to the delay of SP nodes (mentioned below) and the link reliability.
2) *SP Nodes:* The quality of each SP node is the other important factor. SP node may be good, bad, or even malicious. Considering the real situation (most of the SPs intent to be good) and the limitation of space, we exclude the malicious case in the simulation. The *processing delay* and the *processing capacity* are the only two metrics to evaluate the quality of SPs. One unit of *processing capacity* can be used to serve or forward one service request. The negative effects of the shortage of processing capacity are the delay or packet dropping, which can be simulated by the link reliability and the delay of SP nodes. Hence, in the simulation, the quality of each SP node is mainly estimated by the processing delay. Each SP is randomly assigned with a delay factor δ_i. If the delay of the ingress link is D_i, the processing delay of SP is then $D_i * \delta_i$.

From these two angles, we generate the topology of Internet routing with SPs and links with different kinds of qualities.

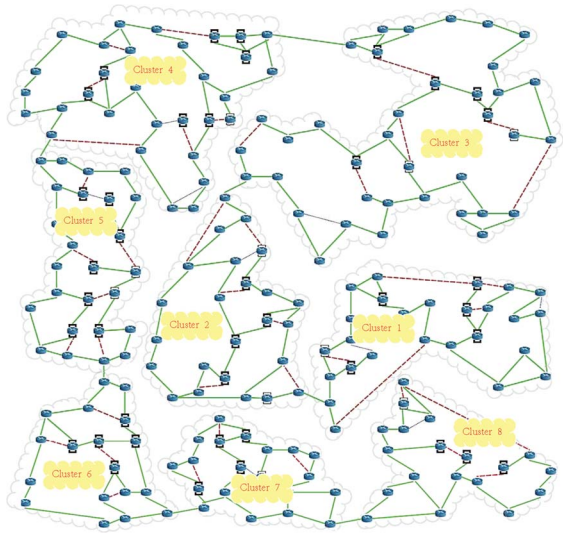


Fig. 4. Topology used in the simulation with eight clusters from one to eight. Each cluster stands for an independent routing area, e.g., a country. The nodes with thick frame, thin frame, and no frame stand for the SPs with high, low, and no delay, respectively. The dashed thick line, the dotted thin line, and the solid thick line represent the links with low, high, and full link reliability, respectively.

Topology generation is a very big part of the simulation code. First topology is generated according to the global minimum and maximum degrees of the SP node (they can be adjusted in the GUI), and the maximum number of the nodes in each specified area. Then, we randomly select some nodes and links to assign different delay and reliability. The topology used in the simulation is shown in Fig. 4, where totally eight connected clusters are generated with labels *Cluster 1* to *Cluster 8*. Each cluster stands for an independent routing area, for example, a country. The red, blue, and green nodes in color mode [gray scale in the black/white (B/W) mode] stand for the SPs with high ($\delta_i = 0.8$), low ($\delta_i = 0.2$), and no delay ($\delta_i = 0$), respectively. The red, blue, and green lines in color mode (gray scale in the B/W mode) represent the links with low ($\Re_{li} = 0.2$), high ($\Re_{li} = 0.8$), and full reliability ($\Re_{li} = 1$), respectively. The details of the parameters in the simulation are shown in Table I. To further study TRECON, in Section V-D, we execute the simulation with parameters under a different topology. The changes of the parameters are shown in Table II.

B. Performance Metrics

To better evaluate the simulation results, we propose six metrics.

1) *Delay Index* $\wp$: Path delay $\wp$ is the sum of the link delay and node delay along the path from the requester to the destination, which is calculated as $\mathcal{D} = \sum(D_i * (1 + \delta_i))$, where D_i is the delay of ith link, and δ_i is the delay factor of ith node in the routing path. In order to integrate to the trustworthiness derivation, it is normalized as $\wp = \mathcal{D}_s/\mathcal{D} = (\sum D_i / \sum(D_i * (1 + \delta_i)))$, where $\mathcal{D}_s$ is the delay of the shortest path without considering the node delay. Therefore, the larger $\wp$ is, the less delay the routing will be. If the routing fails in the middle, then $\wp = 0$.

TABLE I
SIMULATION SETTINGS AND THEIR ILLUSTRATIONS

TRECON related parameters		
Parameters	**Values**	**Illustrations**
ω_d	{0.8, 0.5, 0.2}	Weight of delay
ω_p	{0.8, 0.5, 0.2}	Weight of payment
S_E	10	Size of the environment alert queue
S_H	6	Size of the history list
ρ	0.5	Alert threshold
S_N	Fixed	Number of neighbors
S_R	S_N	Size of the rating queue
Topology related parameters		
Parameters	**Values**	**Illustrations**
N_I	200	Number of SPs
N_L	268	Number of links
U_h	40	Number of highly unreliable links
U_l	10	Number of low unreliable links
D_h	40	Number of highly delayed SPs
D_l	10	Number of low delayed SPs
C_t	200	Total Processing capacity of SPs
Other parameters		
Parameters	**Values**	**Illustrations**
μ	100	Mean of Poison distribution
N_S	3000	Total steps in one simulation

TABLE II
CONFIGURATION FOR NEW GROUP OF SIMULATIONS. ONLY THE CONFIGURATIONS DIFFERENT FROM TABLE I ARE SHOWN

Same Parameters with New Values			
ω_d = 0.2	ω_p = 0.2	N_L = 271	U_h = 18
U_l = 14	D_h = 13	D_l = 3	N_S = 1000

New Parameters	
U_m = 10	Number of medium unreliable links
D_m = 18	Number of medium delayed SPs

2) *Cost for Forwarding Services* ς: The cost for forwarding services is the sum of the payment for asking help from other SPs, and is calculated as $\varsigma = \sum \varsigma_i$, where ς_i is the payment P_{ai} of the ith forward request. In the simulation, ς_i is decided by (1) and (2).
3) *Earn From Forwarding Services* ε: Earn from forwarding services is the sum of profit from helping other SPs to forward the packets, which is calculated as $\varepsilon = \sum \varepsilon_i$, where ε_i is the payment of ith forwarding service. It is the counterpart of ς_i. When the requester pays ς_i for the forwarding service for packet i, the next-hop SP earns $\varepsilon_i = \varsigma_i$ after serving this forwarding service.
4) *Total Request From Clients for SP* i λ_i: λ_i is the total number of requests from clients for SP i.
5) *Net Profits* ρ: ρ is the net profit after subtracting the cost for forwarding services from the sum of profits from clients and the profits from forwarding services, which is calculated as $\rho = U * \lambda + \varepsilon - \varsigma$. U is the unit price for the client request. Profits from end clients are the major profit for the SP. It is reasonable to make the assumption that U is larger than the maximum payment/cost of any one forwarding services, i.e., $\forall i, U > \max(\varsigma_i, \varepsilon_i)$. In the simulation, we set U as 30, and both $\hat{P}_r$ and $\hat{P}_a$ as 20 in (1) and (2) to meet this assumption, and we calculate the profit ρ_i of SP i and the total profit ρ for the whole system as shown in

$$\rho_i = \sum_i (30 * \lambda_i + \varepsilon_i - \varsigma_i) \tag{4}$$

$$\rho = \sum_i \rho_i. \tag{5}$$

TABLE III
SPECIFICATION OF CONFIGURATION C_{ij}

Index of weight of delay i			
Value of i	0	1	2
Value of ω_d	0.8	0.5	0.2
Index of weight of payment j			
Value of j	0	1	2
Value of ω_p	0.8	0.5	0.2
Specification of Configuration C_{ij}			
	i = 0	i = 1	i = 2
j = 0, P > T	D>R	D=R	D<R
j = 1, P = T	D>R	D=R	D<R
j = 2, P < T	D>R	D=R	D<R

6) *Success Rate of Routing* ξ_i: ξ_i is the percentage of successful services to the total number of service requests from clients to the SP i, which is calculated as $\xi_i = \lambda_{si}/\lambda_i$, where λ_{si} is the total number of successful services for all clients (e.g., end-users) of SP i.
7) *Total Traffic* τ_i: τ_i is the total traffic of link i during the whole system running.

Among these metrics, $\wp$ and ξ are the two direct metrics to evaluate the quality of routes. ε, λ, ρ are the metrics to evaluate the economic performance of the TRECON framework; finally τ can be used to evaluate the load balance of the network.

C. Apply aPET in Internet Routing

We need make some modifications to *a*PET in order to apply it to the Internet routing. Since the neighborhood in Internet routing is relatively stable, the neighbor set is fixed. The blacklist is no longer used. 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: $(\wp, \xi_i)$, i.e., we derive the trustworthiness from two quality factors, delay, and success rate. To calculate the trustworthiness value, the rating value R in the formula $T = W * R + (1 - W) * I$ in Fig. 2 changes to $R = (\omega_d * (\Sigma\wp_{iR}) + (1 - \omega_d) * (\Sigma\xi_{iR}))/S_R$, and I changes to $I = (\omega_d * (\Sigma\wp_{iI}) + (1 - \omega_d) * (\Sigma\xi_{iI}))/S_H$, where ω_d is the routing preference weight for the delay and correspondingly $(1 - \omega_d)$ is the weight for the success rate. Note that W is self-adaptive by the *a*PET model. $\wp_{iR}$ and ξ_{iR} are the delay index for neighbor which is the ith element in the rating queue and history table. Similarly, $\wp_{iR}$ and $\wp_{iI}$ are the success rate for neighbor which are the ith element in the rating queue and history table. One complete routing event will incur all the nodes in the route to update the information in the rating queue and history table. Suppose a route is $a \to b \to c \to d$. 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. Let us assume that a wants to send a packet with the routing preference $\omega_d = 0.2$ and $\omega_p = 0.2$, each intermediate SP in the route selects the best next hop based on the value calculated with this preference.

In the simulation, the combination of the weights of delay (ω_d) and payment (ω_p) is called the configuration of one run, which is denoted as C_{ij}. The meaning of C_{ij} is shown in Table III, where i is the index of weight of delay ω_d, and j is the index of weight of payment ω_p. For each value of i and j, there is one corresponding value of ω_d and ω_p. This corresponding

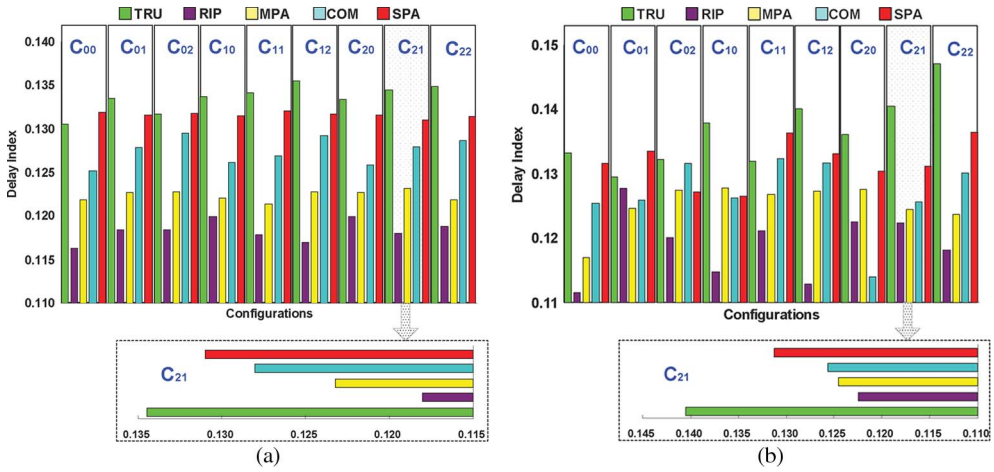


Fig. 5. Comparison of average path delay index $\bar{\wp}$ (a) during the whole system running and (b) during the last 2000 packets routing.

relationship is shown in first two rows in Table III. The last row in Table III shows the mapping in a more direct way, where P, T, D, and R stand for the weight of payment, trustworthiness value, delay, and reliability, respectively, and $>$, $=$, and $<$ stand for the "larger," "equal," and "less" relationship. For example, C_{21} ($i = 2$, $j = 1$) stands for the configuration with $\omega_d = 0.2$ and $\omega_p = 0.5$, which means that the weight of delay is equal to 0.2 (the weight of success rate is correspondingly equal to $1 - 0.2 = 0.8$), and the weight of payment is equal to 0.5 (correspondingly the weight of trust is equal to $1 - 0.5 = 0.5$). Hence, this is a configuration that treats the payment and trustworthiness equally when the *COM* policy is adopted, and puts more weight on delay instead of success rate when deriving the trustworthiness value.

D. Simulation Execution and Data Collection

A round-based simulation is used to test our idea. In each round (step) of the simulation, a certain amount of the service requests are generated with a random SP pair (a, b), in which a is the first SPs in the route (also called access SP), and b is the destination of the routing. The number of the requests generated in each round follows an exponential distribution. To make the data more convincible, we totally conduct 13 groups of simulations (each simulation we call it a run), and each group includes 45 runs. The 45 runs exhaust the combination space of three values of ω_d, three values of ω_p, and five policies. Therefore, totally, we conducted $13 * 45 = 585$ runs. For each parameter combination (same ω_d, ω_p, and policy), we repeat totally 13 runs.

V. Experimental Analysis

Here, we compare five next-hop selection policies in general, *TRU* and *SPA* in particular, from the perspectives of path delay, success delivery rate, link traffic, and economic effects.

A. Analysis of Service Quality

The service quality includes the path delay and the success delivery rate which can be directly measured and are cared about by end users.

1) Path Delay Index ($\wp$): Delay is an important metric to evaluate the routing quality. *SPA* adopts the global shortest path policy, so it is supposed to be the best policy with lowest delay. However, in the real Internet *SPA* is very difficult to fully implement because of the decentralized nature of Internet. Furthermore, *SPA* relies on the link delay to get the shortest path without considering the hidden delays caused by congestion of routers, and the links. On the contrary our trust-related policies, particularly *TRU* select the next hop based on the trustworthiness information, whose value somehow can reflect the delay from the history information. Hence, it is interesting to see whether *SPA* has the best performance in terms of path delay over other four trust-related policies.

The result of the routing delay is shown in Fig. 5. In Fig. 5, we use the path delay index $\wp$, the normalized path delay for the analysis. Fig. 5(a) shows the average delay index $\bar{\wp}$ for all five policies. The results are grouped by different configurations denoted by C_{ij}. Totally there are nine groups, each of which includes five bars corresponding to five policies. The height of the bar is equal to the value of $\bar{\wp}$ for each policy. The group with configuration C_{21} is enlarged and shown at the bottom of Fig. 5(a). The group C_{21} has the configuration as $\omega_d = 0.2$ and $\omega_p = 0.5$, which means that the success rate is more important when deriving the trustworthiness value, and the weight of payment is the same as the trustworthiness value in *COM*. In the left bar charts, the C_{21} group will be again specially enlarged in the bottom. We choose C_{21} as a representative is because the pattern of its results is close to the overall pattern of results from all groups.

From Fig. 5(a), we can easily find that $\bar{\wp}$ is basically decreased following the order $TRU \rightarrow SPA \rightarrow COM \rightarrow MPA \rightarrow RIP$. Since in our simulation the path chosen in *SPA* is the real shortest path, its $\bar{\wp}$ should be largest. According to the definition of delay index, the larger the delay index is, the less the path delay is. Surprisingly, the results show that *TRU* beats *SPA* in all configuration except C_{00}. It shows that *TRU* is good at finding out the effective shortest path by considering the hidden delay and avoiding the unreliable links, while *SPA* considers only link delay. It implies that *TRU* is more applicable and effective than *SPA* in the real routing of Internet. When we look at the details of configuration C_{21} in the bottom of Fig. 5(a), we can see that the results of C_{21}

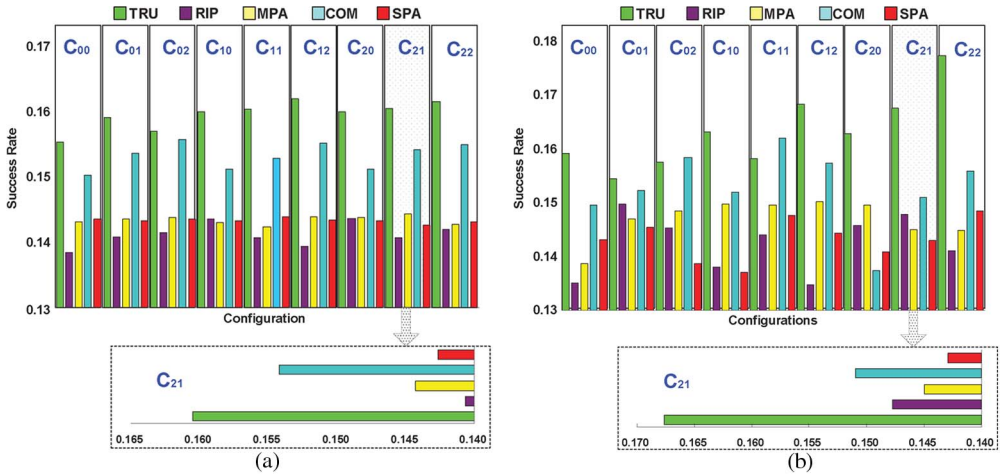


Fig. 6. Comparison of average success rate $\bar{\xi}$ of routing (a) in the whole system running and (b) during the last 2000 packets routing.

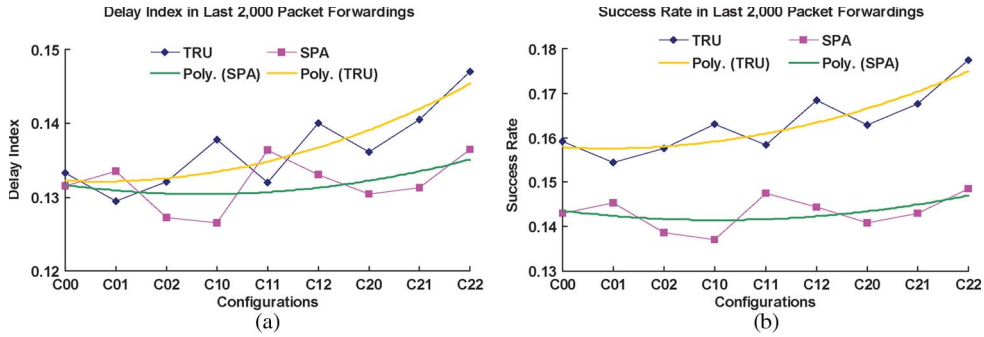


Fig. 7. Comprehensively comparing *TRU* and *SPA* Policies from the angles of (a) $\bar{\wp}$ and (b) $\bar{\xi}$ in the period of last 2000 packet forwarding.

have the same pattern as we find above. We attribute this to the fact that success rate should be more important than delay. Intuitively, the negative effect of routing fail is definitely more than the routing delay. Therefore, putting more weight on the delay in the trustworthiness calculation will degrade the impact of the trustworthiness on next-hop selection. It is also part of the reason why we choose C_{21} which has a larger weight on the success rate, to give the detailed analysis. Observing the result of C_{21} in Fig. 5(a), we can find that $\bar{\wp}$ of *TRU* increases 1.87% comparing with *SPA*. Although the improvement is less than 2%, we are still satisfied because *SPA* is a theoretically optimal routing approach in our simulation. Any improvement on shortening the delay comparing to *SPA* would be treated as a significant improvement. Other three trust-related policies, *COM*, *MPA*, and *RIP* are worse than *SPA*.

Difference from Fig. 5(a) and (b) shows $\bar{\wp}$ in the last 2000 packet routings for each policy. Since the trust-related policies may need time to get converged, the result in the last 2000 packet routings should be more stable and accurate. In Fig. 5(b), we can reach the similar conclusion as in Fig. 5(a), i.e., *TRU* is still the best among all five policies. For the groups C_{12} to C_{22} (weight of success rate is larger or equal to the weight of delay) in Fig. 5(b), we find $\bar{\wp}$ for *TRU* is increased comparing to Fig. 5(a), which confirms our deduction that we should put more weight on success rate in *TRU* to make TRECON work better. In the C_{21} group in Fig. 5(b), $\bar{\wp}$ of *TRU* increases 6.82% from *SPA*. It shows that with a higher weight of success rate on the trustworthiness derivation, *TRU* can show better convergence and performance in reducing the delay.

2) Success Rate of Routing ξ: Next, we compare the success rate of routing of different policies. Similar to the previous section, there are also two figures in Fig. 6 to present the results, where Fig. 6(a) shows the result from the angle of whole run, while Fig. 6(b) shows the result from the angle of last 2000 packet routings. We can see that in both figures, *TRU* outperforms *SPA*. Specially in configuration C_{21} in Fig. 6(a), *TRU* beats *SPA* by 12.6%, while in Fig. 6(b), this percentage increases to 17.2%. Success rate is the direct and the most important metric to show the efficacy of different five policies.

3) Comparing TRU and SPA: Since *TRU* is the policy we advocate in this paper, and *SPA* is the baseline policy, we extract the results of these two policies into separate figures and compare them in more details. In this comparison, we only take the results from the last 2000 packet routings as example, and the results are shown in Fig. 7. We still make the comparison from two angles, i.e., $\bar{\wp}$ and $\bar{\xi}_i$. For each figure in Fig. 7, in order to show the clear comparison, we also add the polynomial regression line for each policy. As shown in Figs. 5 and 6, *TRU* has larger $\bar{\wp}$ and $\bar{\xi}_i$ than *SPA*. We also find that in Fig. 7, basically the polynomial regression lines of *TRU* in figures (a) and (b) are going up from configuration C_{00} to C_{22}, while in (c) the rising trend is not obvious. From configuration C_{00} to C_{22}, the weight of success rate $1-\omega_d$ is increasing from 0.2 to 0.8. Therefore, from polynomial regression lines, we can clear see that increasing the weight of success rate in trustworthiness derivation can improve $\wp$ and ξ of TRECON. The regression lines of *SPA* are quite steady and flat. It meets our expectation: because for each run of *SPA*, the path is fixed, and

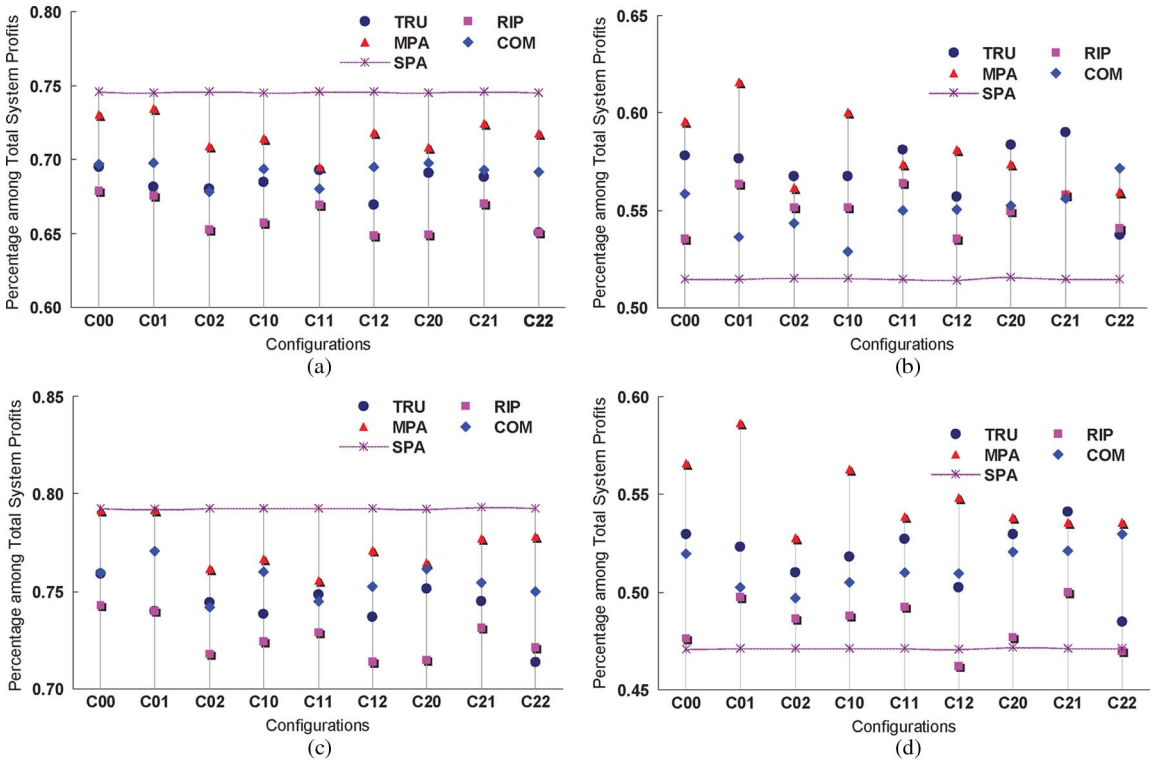


Fig. 8. Economic effects analysis for SP groups with different configurations from the angle of net profit. (a) Profit earned by 150 SPs without node delay ($\wp = 0$). (b) Profit earned by 106 SPs with average link reliability of neighbor links $= 1(\bar{\Re}_l = 1)$. (c) Profit earned by 159 SPs with $\wp \leq 0.5$ and $\bar{\Re}_l \geq 0.5$. (d) Profit earned by 97 SPs with $\wp \leq 0.02$ and $\bar{\Re}_l \geq 0.98$.

the trace is random generated, all results from each run should be close.

4) Discussions: Although the differences of results in the figures are small, we argue that we are still able to draw our conclusions because of the following two reasons.

1) For delay index $\wp = \mathcal{D}_s/\mathcal{D} = (\sum D_i / \sum (D_i * (1 + \delta_i)))$, the more nodes have delay factor δ and the larger of δ is, the smaller the delay index $\wp$ will be. Moreover, if the routing fails in the middle, then $\wp = 0$. Since in the simulation, the number of hops between source and destination is not limited for the random generated routing request, it can be very large. Therefore, there are many chances to meet more delayed nodes or unreliable links which can lower the value of $\wp$. We deduct the main reason to make the value $\bar{\wp}$ to be low is because there is considerable amount of routing with $\wp = 0$. To verify this, we randomly pickup one group experiment with configuration C_{00}, inside which we observe the values of $\bar{\wp}$ and $\tilde{\wp}$. $\tilde{\wp}$ is the average value of all nonzero $\wp$ from successful routing. For *TRU*, these two values are 0.131 and 0.839; for *SPA*, these two values are 0.132 and 0.919, and results from other experiments show the similar pattern. These data validate our deduction. Since we have conducted 13 experiments for each configuration for each policy, and the overall result shows the similar pattern for all nine configurations (C_{00} to C_{22}), we believe our conclusion on these result is reasonable, even though the result difference between 5 policies is small.
2) In the experimental analysis, we choose the *SPA* policy as the baseline. We calculate the shortest path offline, so in the simulation it is the real shortest path, which is different from the approximate shortest path in current Internet. We can expect the performance of *SPA* in the simulation must be better than the approximate *SPA* in the real Internet. Although *SPA* cannot detect the hidden delay of node, it is still the theoretical optimal approach from the angle of link length, and its performance is expected to be close to the best. From this angle, even there is a small improvement compared to *SPA*, we still can claim such an improvement is a significant breakthrough.

To this end, we conclude that *TRU* is better than *SPA* and other three policies in terms of $\wp$ and ξ.

B. Economic Effects Analysis

One of the potential advantages of TRECON is helping the good SPs receive more profits than those low-quality SPs. The profits of one SP are calculated as (4). Fig. 8 shows the percentage of profit among the whole system earned by different groups of SPs. To thoroughly see how the quality of SPs and links affect the economic profit, we choose four representative groups of SPs for study: 1) SPs without any node delay $\wp = 0$; 2) SPs with the average link reliability $\bar{\Re}_l = 1$ (we call them group R_1); 3) SPs with $\wp \leq 0.5$ and $\bar{\Re}_l \geq 0.5$; and 4) SPs with $\wp \leq 0.02$ and $\bar{\Re}_l \geq 0.98$ (we call them group R_1D_0). Group R_1 and R_1D_0 stand for the SPs with high quality. These four groups are shown in Fig. 8(a)–(d), respectively. In all four figures in Fig. 8, the percentage of profits for the four groups of SPs with *SPA* is stable. In Fig. 8(a) and (c), *SPA* is the best policy, and the percentage of profits earned by 75%

TABLE IV
ECONOMIC SPEED RATIO α FOR *MPA*, *TRU*, *COM*, AND *RIP*

Policies		C_{00}	C_{01}	C_{02}	C_{10}	C_{11}	C_{12}	C_{20}	C_{21}	C_{22}
R_1, **53%**	*MPA*	1.124	1.162	1.059	1.133	1.082	1.096	1.082	1.053	1.055
	TRU	1.091	1.088	1.071	1.071	1.096	1.050	1.101	1.114	1.013
	COM	1.054	1.011	1.025	0.998	1.037	1.039	1.042	1.049	1.078
	RIP	1.010	1.063	1.040	1.040	1.064	1.010	1.037	1.053	1.020
R_1D_0, **48.5%**	*MPA*	1.167	1.210	1.088	1.160	1.110	1.131	1.110	1.104	1.104
	TRU	1.092	1.079	1.051	1.068	1.086	1.035	1.092	1.116	1.000
	COM	1.071	1.036	1.024	1.041	1.051	1.050	1.073	1.074	1.092
	RIP	0.982	1.025	1.003	1.006	1.015	0.953	0.983	1.030	0.968

(150) SP nodes and 79.5% (159) SP nodes is 74.7% and 79.0%, respectively. Among all other four trust-related policies, *MPA* is the next best policy. The performance of *TRU* is a little worse than *COM*, and *RIP* is the worst policy for the profit earning. In Fig. 8(b) and (d), we find different pattern. When *SPA* is used, the total profits of group R_1 (53% SPs) and R_1D_0 (48.5% SPs) are only 51.4% and 47.1% in Fig. 8(b) and (d), respectively. In these two groups, *SPA* is almost the worst policy among all five policies. It is even worse than *RIP* which is always worst in the previous analysis; *MPA* then turns to be the best policy in these two groups. *SPA* degrades from the best policy in Fig. 8(a) and (c) to the worst one in Fig. 8(b) and (d), while other trust-related polices, particularly *MPA* have performance upgrading from Fig. 8(a) and (c) to Fig. 8(b) and (d). The reason is in Fig. 8(a) and (c) the percentage of studied SPs reaches to 75% and 79.5%, respectively, among which there are definitely some low-quality SPs inside. Therefore, although the trust-related policies can make the high-quality SPs to earn more profits, the total profits will be traded off because of the low profits of those low-quality SPs. Since *SPA* selects the next hop without considering the SP quality, the profits of each SP is supposed to be approximately the same. Therefore, the more low-quality SPs inside the group, the more total profits of the group for *SPA* can earn than the trust-related policies. However, when the group turns to be a high-quality SP groups like R_1 or R_1D_0, the economic advantage of the trust-related policies shows up.

To better compare the economic effect, we define an economic speedup ratio $\alpha = P_p/P_n$, where P_p is the percentage of the profits among the total profit, and P_n is the percentage of SPs of the group which earns these P_p profits. For a good routing policy, we expect its α value is larger than one. The α values are shown in Table IV. For both SP groups R_1 and R_1D_0, we only list the α values of four trust-related policies, since the α value for *SPA* in both groups are almost the same, i.e., around 0.970 in both cases. Group R_1 and R_1D_0 are the high-quality SP groups. α is expected to be larger when the policies play positively ($\alpha > 1$). However, *SPA*'s α is around 0.970 in both Fig. 8(a) and (c), which means *SPA* is actually playing negatively. Among the other four policies, *MPA* is the best policies: In most configurations in both SP groups R_1 and R_1D_0, it has the largest α value. It indicates that the next-hop selection with lowest payment does help high-quality SPs to increase the profit. However, from the previous results, $\wp$ and ξ of *MPA* are almost the second worst policy among all five policies. High-delay and low-success-rate routing will increase end users' unsatisfaction. Once SPs lose end users, their total profits must decrease from the long run. Therefore, actually, *MPA* is not the best policy we should choose even only considering economic effects. *TRU* is the best policy when considering $\wp$ and ξ. In most of the configurations in both SP groups R_1 and R_1D_0, *TRU* is also the second best policy only worse than *MPA*. In some configurations, for example, C_{21} in both groups, *TRU* (1.114 in R_1, and 1.116 in R_1D_0) can even outperform *MPA* (1.053 in R_1, and 1.104 in R_1D_0) to be the best policy. It shows that *TRU* can be a good alternative of *MPA*. Therefore, when considering all $\wp$, ξ, and the economic incentives, *TRU* is the best policy.

However, we find that in configuration C_{22}, *TRU* has a significant degradation in Fig. 8(b) and (d). In group R_1, its α is only 1.013, which is even worse than *RIP*'s 1.020. Instead, *COM*'s performance has a great promotion in this two figures: $\alpha = 1.078$ and $\alpha = 1.092$ in groups R_1 and R_1D_0, respectively. *COM*'s α in group R_1 is even better than *MPA*'s; even not as high as *MPA*, *COM*'s α in group R_1D_0 is 1.092, very close to *MPA*'s 1.104. It shows that C_{22} is the best configuration in introducing the economic incentives for *COM*. In C_{22}, *COM* derives the trustworthiness with high weight on success rate which make the trustworthiness value to better reflect the true qualities of SPs; when calculating the combination value of trustworthiness and payment, as shown in (3), *COM* has low weight on payment. With this combination, *COM* can better integrate the trustworthiness and payment factors together to make a good routing selection. As we have found in the previous analysis, *COM* also performs well in reducing the delay and increasing the success rate in configuration C_{22}.

In summary, although *MPA* is the best policy in the economic incentives introducing, it is not good in improving the service quality. *TRU* is not as good as *MPA* from the perspective of economy, but it is the best policy to provide good services. *COM* is in the middle of *MPA* and *TRU*. It is worse than *TRU* in providing good services, but it has better economic effects. These observations imply that a possible direction for further improvement of our approach, i.e., the improvement should keep the advantages in *TRU* in providing more good services, as well as economic advantage of *MPA* from the perspective of economic effects. Integrating our *M-CUBE* model into TRECON is going with this direction. However, carefully designing the configuration for *COM* may also be another future improvement option.

C. Comparison of Different Network Settings

So far, all the results we have obtained in the previous analysis are from the simulations using the configurations as shown Table I (denoted as C_{old}). From Table I, we can find that there are totally 20% (40) high-delayed and 5% (10) low-delayed SPs, and 14.93% (40) highly unreliable and 3.73% (10) low unreliable links. We choose these exaggerated (worse) configurations intentionally in order to show the effect of our framework. However, we envision that the real network could

TABLE V
COMPARE THE RESULT UNDER TWO CONFIGURATIONS: C_{ij_old} AND C_{ij_new} ARE THE CONFIGURATION IN THE OLD AND NEW SIMULATIONS, RESPECTIVELY

Comparison of Path Delay Index $\wp$						
Policies	Average $\wp$ in the whole system running			Average $\wp$ in the routings of last 2,000 packets		
	C1 (= C_{00_new})	C2(= C_{00_old})	C3 (= C_{20_old})	C1 (= C_{00_new})	C2(= C_{00_old})	C3 (= C_{20_old})
TRU	0.512	0.135	0.137	0.526	0.120	0.150
RIP	0.342	0.119	0.117	0.355	0.118	0.111
MPA	0.341	0.121	0.124	0.329	0.119	0.131
COM	0.317	0.132	0.130	0.337	0.141	0.127
SPA	0.355	0.131	0.132	0.341	0.126	0.130
Comparison of Success Rate of Delivery ξ						
Policies	Average ξ in the whole system running			Average ξ_i in the routing of last 2,000 packets		
	C1 (= C_{00_new})	C2(= C_{00_old})	C3 (= C_{20_old})	C1 (= C_{00_new})	C2(= C_{00_old})	C3 (= C_{20_old})
TRU	0.551	0.162	0.164	0.564	0.144	0.179
RIP	0.427	0.142	0.140	0.449	0.142	0.134
MPA	0.414	0.142	0.145	0.402	0.140	0.155
COM	0.392	0.158	0.156	0.418	0.168	0.151
SPA	0.438	0.143	0.143	0.429	0.138	0.142

have a better configuration. In this section, we conduct another group of simulations in order to find out the behavior (trend) of TRECON in different settings. The new configurations (denoted as C_{new}) are shown in Table II, which represents a healthier SP network than the one simulated with configurations in Table I. We compare the results from the old and new configurations with two metrics: $\wp$ and ξ. In C_{new}, the percentage of high and low unreliable links is only 9% (18) and 7% (14), respectively, instead of 20% and 5% in C_{old}. However, we introduce 5% (10) medium $(0.3 \leq \Re_i < 0.7)$ unreliable links in C_{new}. Regarding the SP nodes, the percentage of high- and low-delayed SP is only 6.5% (13) and 1.5% (3), respectively, in C_{new}, instead of 20% and 5% in C_{old}. However, we introduce 9% (10) medium $(0.4 \leq \Re_i < 0.7)$ delayed SPs in C_{new}.

Because of the space limitation, we only compare three typical cases, $\text{C1} = C_{00\_\text{new}}$ from C_{new}, and $\text{C2} = C_{00\_\text{old}}$ and $\text{C3} = C_{20\_\text{old}}$ from C_{old}. From Table V, we can easily tell that for both metrics, the values of $\wp$ and ξ, in both *the whole system running* and the *last* 2000 *package routings* in C1 are much larger than those in C2 and C3. In particular, for the comparison of *TRU* and *SPA* in C1, the improvement is 44.2% and 25.8% for $\wp$ and ξ, respectively, in the whole system running; while in the last 2000 packages routings, these two numbers increase to 54.3% and 31.5%. The advantages of *TRU* over *SPA* are significantly more than the cases of C2 and C3. It shows that *TRU* can show more advantages in a healthier environment; when the environment turns worse (more high-delayed SPs and highly unreliable links), the advantages of *TRU* are reduced.

Although *TRU*'s advantages in C3 are still not so good as the ones in C1, when we turn to the last 2000 package routing routings, we find that *TRU*'s advantages in C3 are obviously better than C2. More specifically, when we increase the weight of the success rate $(1 - \omega_d)$ from 0.2 (C2) to 0.8 (C3), the advantages of *TRU* are improved for $\wp$, $\Re$, and ξ. It further confirms what we have discussed before, i.e., increasing the weight of success rate in the trustworthiness derivation can make the trust model more effective.

D. Delay Index and Success Rate Under New Topology

We have presented the running results under the topology shown in Fig. 4. However, are these results consistent under another topology, particularly for *TRU* that we are advocating? To answer this question, we conduct the last group of experiments under a new topology, as shown in Fig. 9. Comparing with Fig. 4, most parameters in new topology are the same in Table I, except the total number of links and the distribution of low-quality nodes and links in each cluster. Because the time issue, we only conduct two groups of simulations with total 90 runs, i.e., for each parameter combination (the same ω_d, ω_p, and policy), we repeat totally two runs. Due to the space limit, we choose the most important results about average path delay index $\bar{\wp}$ and average success rate $\bar{\xi}$ to present. The results are shown in Fig. 10. Compared with the results in Figs. 5(a) and 6(a), there are two main differences in the results: 1) Both $\bar{\wp}$ and $\bar{\xi}$ significantly increase. 2) Except *TRU*, the result pattern of the other four policies have some variations. The main reasons to incur this difference should be the change of the topology.

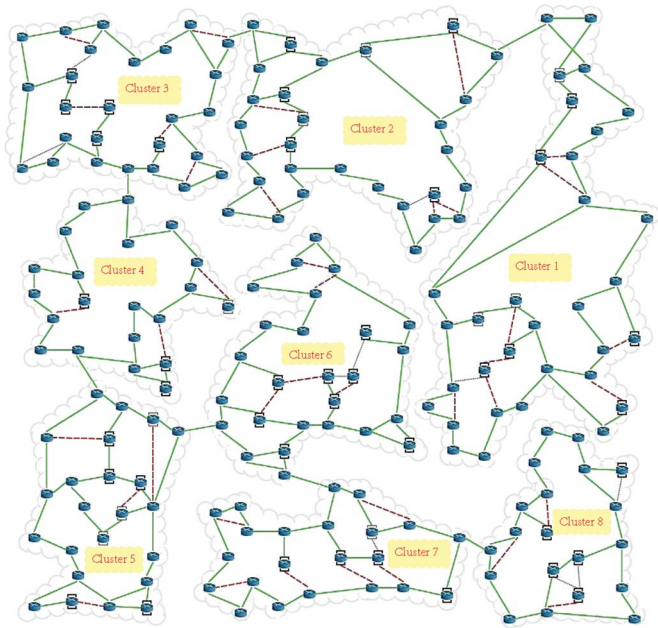


Fig. 9. New topology with eight clusters.

Another reason may be because only two runs are conducted for each parameter combination, so that the results are not as stable and confident with the results in Figs. 5(a) and 6(a).

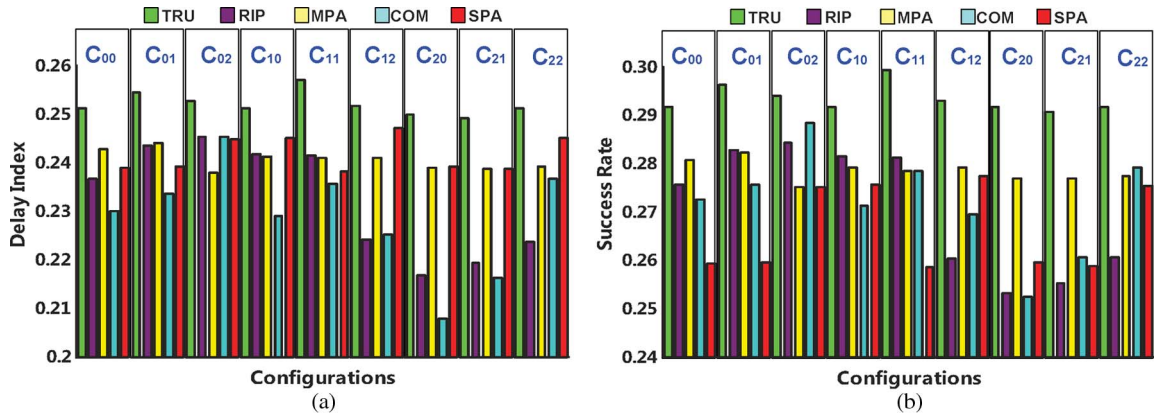


Fig. 10. (a) Average path delay index $\bar{\wp}$ and (b) the average success rate $\bar{\xi}$ under new topology, both are during the whole system running.

Although there are these differences, *TRU* is still the best one in Fig. 10 for both $\bar{\wp}$ and $\bar{\xi}$, and its advantage over other policies are even more distinguished and consistent. From this, we further confirm that *TRU* is effective.

E. Summary

We have comprehensively evaluated five routing policies. From all the results, we conclude that *TRU* is the best policy when we consider both the routing performance and economic effects. Specifically, we observe the following: 1) *TRU* is the best policy from the angle of reducing the routing delay and increasing the routing success rate. Compared with *SPA*, *TRU* can also significantly reduce the link traffic. From the economic angle, *TRU* is only slightly worse than the economic-intensive policy *MPA*. 2) Although *MPA* is best in introducing positive economic effects into the routing, its performance on improving the routing quality is poor. 3) By integrating *TRU* and *MPA*, *COM* cannot show obvious advantages with improving both the routing quality and economic effects. It is worse than just using *TRU* in most of the cases which shows that the combination of *TRU* and *MPA* is not a good option, or at least it needs a better way other than simply weighted sum for the combination. 4) Although *SPA* is an optimal approach, it is completely beaten by *TRU* due to its incapability to catch the hidden delay and the quality of links and SPs, and to balance the link traffic, and poor performance from the economic angles. From above, we can argue that TRECON using *TRU* is promising in the design of next-generation Internet, and it is a good alternative for current IGP and BGP.

VI. Related Work

This paper builds upon two categories of previous efforts: 1) Internet routing and 2) trust and reputation models.

A. Internet Routing

BGP [12] has several problems like the oversized routing table and deficiencies in the scalability.

Due to the deficiency of current BGP protocol, researchers have taken steps to build the next-generation Internet. In [18], Siganos and Faloutsos develop a methodology and tool for interfacing and cross-comparing the two major sources of BGP policy information: Internet routing registry at the configuration plane and BGP routing tables at the operation plane. HLP [13] aims to more scalability, better isolation, and faster convergence than the current BGP routing for the next-generation interdomain routing. While HLP focuses more on the adaptability to different policies, TRECON focuses on the concrete policy design based on the trustworthiness and economic effects. AIP [19] addresses the problem from a different angle. It seeks a simple change to Internet addressing to allow the Internet routing to achieve goals of accurate reflection of network-layer reachability, secure routing messages, and effective traffic control. AIP complements to this paper very well. `Nexit` [20], a negotiation framework to support negotiation-based routing between neighboring SPs, is similar to this paper in the policy building. Both `Nexit` and TRECON try to build the collaboration routing between competing entities. However, `Nexit` relies on the negotiation to reach the agreement on the path selection; in TRECON, each SP makes the decision independently based on the available trustworthiness and price information. Different with `Nexit`, FBR [21] makes the path selection in the routing based on feedbacks, which is close to this paper from the angle of routing policy building. However, in TRECON, not only the feedbacks (ratings) but also the self-experiences of SPs are integrated to make the routing decision. Reference [22] proposes soft preemption to reroutes a connection before it is tore down so that the interruption of ongoing service can be avoided. Soft preemption can be used in TRECON to dynamic rebuild the high-quality routes.

B. Trust and Reputation Models

Trust inference or reputation-based systems has been a hot topic and studied in the literature [6], [8], [9], [23]–[30].

The notion of "trust management" was first coined by Blaze *et al.* [31] in their seminal paper on decentralized trust management. Marsh [32] is the first one to introduce a computational model for trust in the distributed artificial intelligence (DAI) community. Some researchers [25], [27], [28] suggest the mathematical definition from the point of probability and uncertainty. In this paper, similar as Mui *et al.* [8], we choose the term "subjective expectation" rather than "subjective probability" to emphasize that trust is a summary quantity that an entity has toward another based on the historical interactive information between them of trust and reputation.

Centralized reputation system has been widely deployed in e-commerce [6], [23], [33], such as eBay (an online auction site) and slashdot.com (an online tech-guru site). Recently, in the P2P domain many decentralized reputation management schemes emerge. Kamvar *et al.* [34] present EigenTrust, a distributed and secure method to compute global trust values based on "Power Iteration." Peers ask their acquaintances about their opinions about other peers to know about other peers. EigenTrust assumes there are pretrusted nodes in the system, which is not applicable in distributed open systems. NICE project [35] discusses the trust inference problems. Zhou and Hwang [30] propose PowerTrust that leverages the power-law feedback characteristics to derive a global trustworthiness value for each peer. Srivatsa *et al.* [26] propose TrustGuard, a trust model where the trustworthiness value is derived from the ratings, and the rating is aggregated with the weighted sum based on the similarity of the rater's experience. These approaches [26], [30], [34] share the same goal of finding a global trustworthiness value for each peer, using different distributed approaches. Different with them, our *a*PET model is a personalized trust model where there is no global trustworthiness value existing in the system.

VII. Conclusion

By observing that the drawbacks of current Internet routing not only include the routing deficiencies but also lack the embedded economic mechanisms to coordinate the SPs, in this paper, we propose a trust-based economic framework TRECON to attack these open problems in Internet routing for the future Internet. The evaluation shows that *TRU* has significant advantage over other policies in reducing the delay, increasing the success rate, balancing the load among SPs and links, and assuring the major profits flow to good SPs. In summary, we believe the proposed TRECON framework is a promising approach by which an accountable future Internet can be built.

A. Outstanding Challenges

The approach proposed in this paper is offered as a starting point for debates about the future Internet routing design. There are some outstanding challenges for the future work.

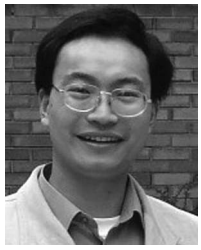
1) We have validated our approach with comprehensive simulation. Based on the simulation results, we will move further to formalize our model and prove its validity with the mathematical analysis.
2) QoS support has been considered important in the future Internet. Our approach has built-in QoS support; however, many other work needs to be done, including specification, protocol design, etc.
3) We will also focus on developing a more mature and practical economic model, and an effective and general clustering algorithm.

References

[1] *Sina.com Report on Taiwan Earthquake*. [Online]. Available: http://tech.sina.com.cn/i/2007-01-05/09131320398.shtml
[2] *A Brief History of Internet*. [Online]. Available: http://www.walthowe.com/navnet/history.html
[3] *Report of NSF Workshop on Overcoming Barriers to Disruptive Innovation in Networking*, Jan. 2005. [Online]. Available: http://planet-lab.org/doc/barriers.pdf
[4] L. Subramanian, V. Padmanabhan, and R. Katz, "Geographic properties of Internet routing," in *Proc. USENIX Annu. Tech. Conf.*, Monterey, CA, Jun. 2002, pp. 243–259.
[5] U. Wilensky, *Netlogo*. Evanston, IL: Northwestern Univ., 1999.
[6] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems," *Commun. ACM*, vol. 43, no. 12, pp. 45–48, Dec. 2001.
[7] A. Jøsang, E. Gray, and M. Kinateder, "Analysing topologies of transitive trust," in *Proc. FAST*, Pisa, Italy, Sep. 2003, pp. 9–22.
[8] L. Mui, M. Mohtashemi, and A. Halberstadt, "A computational model of trust and reputation," in *Proc. HICSS-35*, 2002, p. 188.
[9] Z. Liang and W. Shi, "PET: A PErsonalized Trust model with reputation and risk evaluation for P2P resource sharing," in *Proc. HICSS-38*, Jan. 2005, p. 201.2.
[10] Z. Liang and W. Shi, "Analysis of ratings on trust inference in open environments," in *Perform. Eval.*, Feb. 2008, vol. 65, no. 2, pp. 99–128.
[11] Z. Liang and W. Shi, "Enforcing cooperative resource sharing in untrusted peer-to-peer environment," in *ACM J. Mobile Netw. Appl. (MONET)*, Dec. 2005, vol. 10, no. 6, pp. 771–783.
[12] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (bgp-4)*, Mar. 1995. [Online]. Available: http://www.faqs.org/rfcs/rfc1771.html
[13] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica, "HLP: A next-generation interdomain routing protocol," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 13–24.
[14] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 1371–1380.
[15] B. Cheswick, H. Burch, and S. Branigan, "Mapping and visualizing the Internet," in *Proc. USENIX Annu. Tech. Conf.*, Jun. 2000, pp. 1–12.
[16] N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for network engineering," in *Proc. ACM SIGMETRICS*, Jun. 2004, pp. 331–342.
[17] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz, "Stable egress route selection for interdomain traffic engineering: Model and analysis," in *Proc. ICNP*, Boston, MA, Oct. 2005, pp. 16–29.
[18] G. Siganos and M. Faloutsos, "Analyzing BGP policies: Methodology and tool," in *Proc. IEEE INFOCOM*, Hong Kong, 2004, pp. 1640–1651.
[19] M. Vutukuru, N. Feamster, M. Walfish, H. Balakrishnan, and S. Shenker, Revisiting Internet Addressing: Back to the Future! [Online]. Available: cs.shenker.net/files/aipposn.pdf
[20] R. Mahajan, D. Wetherall, and T. Anderson, "Negotiation-based routing between neighboring ISPs," in *Proc. NSDI*, May 2005, pp. 29–42.
[21] D. Zhu, M. Gritter, and D. Cheriton, "Feedback based routing," in *Proc. HotNets-I*, Oct. 2002, vol. 33, pp. 71–76.
[22] C. H. Lau, B. Soong, and S. K. Bose, "Preemption with rerouting to minimize service disruption in connection-oriented networks," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 5, pp. 1093–1104, Sep. 2008.
[23] K. Aberer and Z. Despotovic, "Managing trust in a peer-to-peer information systems," in *Proc. CIKM*, Nov. 2001, pp. 310–317.
[24] Z. Despotovic and K. Aberer, "P2P reputation management: Probabilistic estimation vs. social networks," *Comput. Netw.*, vol. 50, no. 4, pp. 485–500, Mar. 2006.
[25] A. Jøsang, "Abductive reasoning with uncertainty," in *Proc. IPMU*, Jun. 2008, pp. 9–16.
[26] M. Srivatsa, L. Xiong, and L. Liu, "TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks," in *Proc. WWW*, 2005, pp. 422–431.
[27] Y. Wang and M. P. Singh, "Trust representation and aggregation in a distributed agent system," in *Proc. AAAI*, 2006, pp. 1425–1430.
[28] Y. Wang and M. P. Singh, "Formal trust model for multiagent systems," in *Proc. AAAI*, 2007, pp. 1551–1556.
[29] B. Yu and M. P. Singh, "Searching social networks," in *Proc. AAMAS*, Melbourne, Australia, Jul. 2003, pp. 65–72.
[30] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 460–473, Apr. 2007.
[31] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. IEEE Symp. Security Privacy*, May 1996, p. 164.
[32] S. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, Univ. Stirling, Stirling, U.K., 1994.
[33] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Providing justifications in recommender systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1262–1272, Nov. 2008.
[34] S. Kamvar, M. T. Schlosser, and H. Gacia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proc. WWW*, May 2003, pp. 640–651.
[35] S. Lee, R. Sherwood, and B. Bhattacharjee, "Cooperative peer groups in nice," in *Proc. INFOCOM*, Mar. 2003, vol. 2, pp. 1272–1282.

Zhengqiang Liang received the B.S. and M.S. degrees in computer science and engineering from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively. He is currently working toward the Ph.D. degree in computer science at Wayne State University, Detroit, MI.

His research focuses on trusted and cooperative resource sharing in the open environment, trust-based resource scheduling in open scientific discovery infrastructure, next-generation Internet, peer-to-peer systems, and computer economics.



Weisong Shi (SM'09) received the B.S. degree in computer engineering from Xidian University, Xi'an, China, in 1995 and the Ph.D. degree in computer engineering from the Chinese Academy of Sciences, Beijing, China, in 2000.

He is currently an Associate Professor of computer science with Wayne State University, Detroit, MI. His current research focuses on high-performance computing, distributed systems, and mobile computing. He is the author of the book *Performance Optimization of Software Distributed Shared Memory Systems.*

Dr. Shi is a recipient of a Microsoft Fellowship in 1999, the President outstanding award of the Chinese Academy of Sciences in 2000, one of the 100 outstanding Ph.D. dissertations (China) in 2002, "Faculty Research Award" of Wayne State University in 2004 and 2005, and the "Best Paper Award" of ICWE'04 and IEEE IPDPS'05. He is a recipient of a National Science Foundation CAREER Award in 2007.