Distributed Pagerank:
A Distributed Reputation Model for Open Peer-to-Peer Networks

Atsushi Yamamoto†, Daisuke Asahara††, Tomoko Itao†, Satoshi Tanaka†, and Tatsuya Suda† † †
†NTT Network Innovation Laboratories
3-9-11 Midori-cho, Musashino-shi, 180-8585 Japan
{yamamoto, tomoko, satoshi}@ma.onlab.ntt.co.jp
††Department of Applied Mathematics and Physics,
Graduate School of Informatics, Kyoto University
Kyoto-shi, 606-8501 Japan
asahara@amp.i.kyoto-u.ac.jp
† † †University of California, Irvine
Irvine, CA 92697-3425, USA
suda@ics.uci.edu

Abstract

This paper proposes a distributed reputation model for open peer-to-peer networks called distributed pagerank. This model is motivated by the observation that although pagerank has already satisfied the requirements of reputation models, the centralized calculation of pagerank is incompatible with peer-to-peer networks. Distributed pagerank is a decentralized approach for calculating the pagerank of each peer by its reputation, in which the relationship between peers is introduced as the equivalent to the link between web pages. The distributed calculation of pagerank is performed asynchronously by each peer as it communicates with the other peers. The asynchronous calculation accomplishes both demanding no extra messages for the calculation of pagerank and steadily calculating an accurate pagerank of each peer even under the dynamic topology of relationships. The result of the simulation has indicated that the calculated pagerank value of each peer converges at the original pagerank value under the static topology of relationships, which is presumable under a dynamic topology. A fully implemented application of distributed pagerank has also been presented, which supports dynamic formation of communities with reputation ranking.

1. Introduction

In the near future, a world where all objects are openly connected through a universal network will become a reality. Objects interact with one another and collectively provide services. Open peer-to-peer networks are emerging as probable architectures for such a future world. In using open peer-to-peer networks as commerce, there are still a number of issues to be resolved, including those related to connectivity and reliability. The establishment of reputation models is a matter of key reliability issues affecting the overall risk of communications. Reputation models are required for obtaining an objective reputation of each peer from a global viewpoint of the peer group, which makes it possible to evaluate peers that are not known to all peers. The objective reputation should be reflected appropriately by a subjective one from a local viewpoint of each peer. In the reflection, it is essential to take into consideration the dependence among peers and the resistance to malicious peers.

K. Terada and T. Araragi [1] proposed a reputation model with a global objective reputation of each peer. In this model, the local subjective reputation of each peer may be taken into its calculation. However, this model assumes that the relationship among peers is independent, and has no means to cope with spiteful peers. On the other hand, R. Chen and W. Yeager [2] proposed a reputation model with a local objective reputation of each peer. In this model, the local objective reputation is calculated indirectly based on a local subjective reputation of the object that each peer owns. However, this model does not mention the global objective reputation of each peer.

This paper proposes a distributed reputation model for open peer-to-peer networks called distributed pagerank.

Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW'04)
0-7695-2050-2/04 $20.00 © 2004 IEEE

This model is motivated by the observation that although pagerank has already satisfied the requirements of reputation models, the centralized calculation of pagerank is incompatible with peer-to-peer networks [3]. Distributed pagerank is a decentralized approach for calculating the pagerank of each peer by its reputation, in which the relationship between peers is introduced as equivalent to the link between web pages. In distributed pagerank, instead of a centralized calculation server with global information about the link structure, each peer with local information about the relationship structure calculates its own global pagerank in communicating with other peers asynchronously. As a result of the asynchronous calculation, there is no need for extra messages to calculate pagerank. Because the asynchronous calculation enables continuous updating of pagerank, it can offer an accurate pagerank of each peer even when relationships between peers are created and deleted, or peers with relationships enter and leave dynamically.

The rest of this paper is structured as follows: Section 2 introduces the algorithm for calculating distributed pagerank. Section 3 provides a simulation of a proposed distributed pagerank. Section 4 describes an application of distributed pagerank, and Section 5 concludes this paper.

2. Distributed pagerank

We explain original form of pagerank, and then propose distributed pagerank in this section.

2.1. Original form of pagerank

2.1.1. Overview

L. Page et al. [3] developed pagerank to measure the pagerank value of web pages. They gave an intuitive description of pagerank; a page has a high rank if the sum of the ranks of its backlinks is high. This covers both the case when a page has many backlinks and when a page has a few highly ranked backlinks. The pagerank value of a web page i, denoted $PR(i)$, is equivalent to the number of users on page i after infinite steps of a random walk which proceeds at each step as follows. Users move uniformly at random with probability $(1-d)$ to one of the pages linked from the current page, and users jump with probability d to some other page at random without tracing the linked pages. The value of pagerank is defined recursively according to the equation,

$$PR(i) = d + (1-d)\sum_{j\rightarrow i}\frac{PR(j)}{N(j)}\,, \qquad (1)$$

in which the sum is taken over all web pages j which have a link to page i, such that $N(j)$ is the total number of links originating from page j, where d is a number between 0 and 1. Intuitively, the value of $PR(i)$ expresses the relevancy of the web page i. This ranking is used as one factor of the Google search engine for determining how to order the pages returned by a web search query [4]. In Google, robots crawl the web pages, and then store their information into their database to calculate the pagerank value. Therefore, Google is characterized as a centralized system.

2.1.2. Features

The following pagerank features are considered to satisfy the requirements of a reputation model.

- One is that the value calculated by pagerank is global, i.e., commonly available for any peer in a network.
- The other is that pagerank is based on the permission of all the peers, such as which peers can be chosen as relationship partners and which peers can be removed from being relationship partners.

Note that no peer is able to earn a high pagerank value by cheating the pagerank system since relationships from valuable peers are required to get a high pagerank value, in which valuable peers mean peers who have a positive pagerank value.

2.2. Proposed distributed pagerank algorithm

2.2.1. Approach

To apply pagerank to our reputation model, we develop a new algorithm, named distributed pagerank, to calculate the pagerank value. In our scheme, peers which have received a message perform the procedure described in Section 2.2.2, and relay a message with the updated value to another peer. Note that this is performed without synchronizing with all the other network peers.

Distributed pagerank requires neither any centralized calculation server nor a crawling robot. In this respect, distributed pagerank is different from a web search engine (centralized system). The proposed calculation system does not require any special messages made only for this calculation (we utilize communication messages used for some other purposes between peers). In this point, ours is different from other related work, i.e., ours is considered to be more efficient for network traffic [5].

We present a distributed pagerank algorithm that is able to adjust to a dynamic environment more effectively than the original pagerank algorithm, as another good aspect of distributed pagerank. This is because our algorithm is performed in a successive and purely distributed way although the original pagerank needs the crawling of robots and the analysis of a whole network.

We assume the case that d is zero in the equality (1) since the way to reach peers is only by tracing relationships in our scenario.

2.2.2. Mechanism

To perform distributed pagerank, we define relationship as a directed logical connection link. Note that peers do not necessarily have relationships with each other. Next, we assume each peer has two tables, such as the outlink and inlink tables. The outlink table of peer X indicates the values assigned to each of X's relationship partners. All the values on outlink table are the same as $PR(\text{X})$ divided by the number of X's relationships. X's inlink table indicates the values from all the peers Y that have a relationship to X. The value for Y on X's inlink table is the value conveyed by the latest message from Y. $PR(\text{X})$ is calculated as the sum of all the values on X's inlink table.

Next, we explain the procedure of distributed pagerank, which enables each peer i to automatically know $PR(i)$.

1. When peer A makes a message,

- **(1)** peer A selects one or more peer(s) B out of A's partners (the way to select peers depends on the message type), and then,
- **(2)** marking up the value corresponding to B on A's outlink table in a message, and forwarding it to B (note that no other partner except for B knows the value on A's outlink table).

2. When a peer receives a message, it performs either case (a) or case (b) depending on the situations.

case (a) If this peer is the one whom the originating peer wants to communicate with, the peer performs the procedure "table update," which includes the following three procedures: updating its inlink table according to the value conveyed by a message, calculating its own pagerank based on its inlink table, and updating its outlink table. Then, the peer finishes the message processing.

case (b) Otherwise, this peer completes "table update," forwarding this message to one or more peer(s) with the updated value.

3. Simulation

To demonstrate the effectiveness of distributed pagerank, we ran a simulation described in this section.

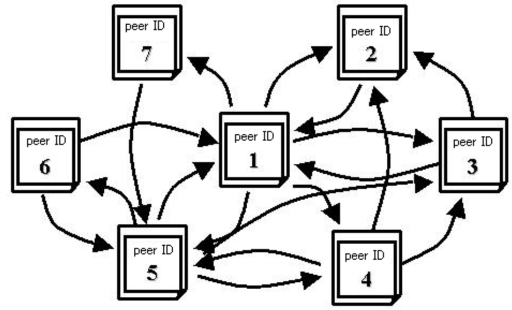

Figure 1. Simulation environment.

3.1. Model

Consider the following environment, Figure 1. Each square represents a peer, and an arrow from peer i to peer j indicates that peer i has a relationship to peer j.

We set $d = 0$ in equality (1), and apply this equality to the relationship structure in Figure 1 for all peers i $(i = 1, 2, \ldots, 7)$. Adding the normalization condition, $\sum_{i=1}^{7} PR(i) = 1$, we get pagerank values as below,

$$\begin{pmatrix} PR(1) \\ PR(2) \\ PR(3) \\ PR(4) \\ PR(5) \\ PR(6) \\ PR(7) \end{pmatrix} = \begin{pmatrix} 0.303514\cdots \\ 0.166134\cdots \\ 0.140575\cdots \\ 0.105431\cdots \\ 0.178914\cdots \\ 0.044728\cdots \\ 0.060703\cdots \end{pmatrix}. \quad (2)$$

We assume the situation in which a peer sends a message to another peer to communicate. We utilize this message to convey updated values between peers. The simulation details are as below:

1. Each peer has an initial pagerank value of 1, making its outlink table. To make an initial inlink table, let each peer get the values from corresponding peers' outlink table. After that, each peer performs "table update." These operations define initial values of each peers' inlink table, outlink table, and pagerank.

2. Peer 1 starts the first communication using a single message at time 0. No peer can start the communications except for case (a). Peer 1 chooses peer X from among its partners, forwarding a message with the value corresponding to X in the outlink table of peer 1. Each peer receiving a message, such as peer X, follows either case (a) with probability p, or case (b) with probability $(1 - p)$.

Table 1. Pagerank value at time 515.

	before normalization	after normalization
PR(1)	2.351770	0.303514
PR(2)	1.287284	0.166134
PR(3)	1.089241	0.140575
PR(4)	0.816930	0.105431
PR(5)	1.386306	0.178914
PR(6)	0.346577	0.044728
PR(7)	0.470354	0.060703

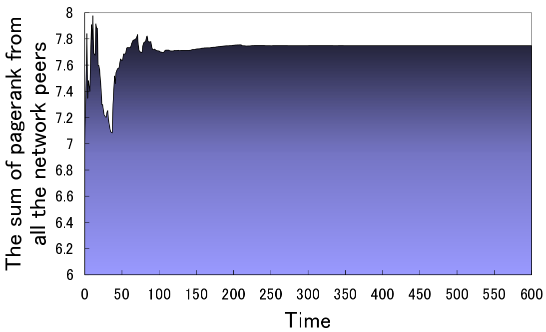

Figure 2. Total sum of pagerank.

case (a) Peer X is the one whom the originator peer 1 wants to communicate with. Peer X performs "table update," and finishes this message processing. In the next place, we randomly choose a peer from among all the network peers, which starts the next new communication in the same way as the originator peer 1 acted in the first communication.

case (b) Peer X is not the one whom the originator peer 1 wants to communicate with. The "table update" procedure is completed. In the next step, peer X chooses a peer from among its partners with equal probability, forwarding a message with the updated value on the outlink table of peer X.

3.2. Results

We performed a simulation assuming that discrete time 1 is taken for a message to hop between peers through relationships. The typical simulation result $p = 0.2$ is shown in the following. Table 1 expresses $PR(i)$ $(i = 1, 2, \ldots, 7)$ and the vertical scale of Figure 2 expresses the time series sum of the pagerank value that all the network peers have.

From Table 1, we see the pagerank value of each peer converges to the original pagerank value denoted in (2) after normalization. From Figure 2, we see the sum of pagerank value is static after each peer's pagerank value has converged. We confirmed these features in every performed simulation including the ones performed by setting different initial values.

We consider how distributed pagerank adapts to a dynamic relationship environment, i.e., whether distributed pagerank converges, or not, to the original pagerank in a dynamic relationship environment. The convergence holds if enough time passes after relationships are changed. Otherwise, the value of distributed pagerank gets close to the value of the original pagerank. This is because we can see a time point when relationships are changed as a renewal point from the observation that the convergence holds independently of the initial pagerank values.

4. An application of distributed pagerank

This section describes an application of distributed pagerank called Meet@. Meet@ is for supporting the dynamic formation of communities. Distributed pagerank is used as a tool to rank members of a community. This allows each user to judge whether unknown members are reputable in the community. Meet@ is implemented using Ja-Net, which is an architecture for open peer-to-peer networks [6].

4.1. Concept of communities formation

Since Meet@ treats users as peers, it can be considered that the relationship between peers is a human relationship, which is created and deleted, the strength of which is changed dynamically. The application programming interface (API) of Ja-Net is suitable for the dynamic characteristic of these relationships [7].

The collection of links of relationships between peers is regarded as a community. Meet@ is for supporting the dynamic formation of communities that reflect local interactions among users in real world environments or spaces. In Meet@, a community is spontaneously formed based on the history of communications among users who happened to visit the same place and communicate with one another. The community is adaptively reformed based on explicit evaluations among users so that popular users (i.e., users who are preferred by many other users) remain in the community while not-so-popular users are removed from the community.

4.2. Dynamic formation of communities

Meet@ assumes a wireless LAN environment. The area covered by a wireless LAN access point is defined as a physical space. Meet@ provides a function to search for

Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW'04)
0-7695-2050-2/04 $20.00 © 2004 **IEEE**

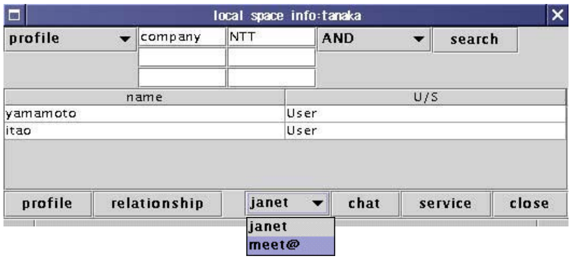

Figure 3. Screenshot of physical space search.

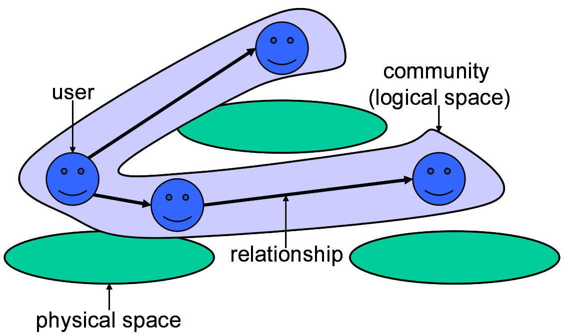

Figure 4. Relation between physical space and community.

users sharing the same physical space (see Figure 3). A relationship is created between two users in the following manner. Users can create a new relation name, which is an attribute of a relationship, such as "movie," or select an existing relation name from the user interface of Meet@. This enables users to enter text-based chat and communicate with their partners in a relationship. In Meet@, a network of relationships with the same relation name is defined as a community (see Figure 4). Relationships are unidirectional. This results in an asymmetric community. Meet@ provides a type of search which follows relationships with an arbitrary relation name among users with respect to oneself (see Figure 5). This permits each user to find members

Figure 5. Screenshot of community search.

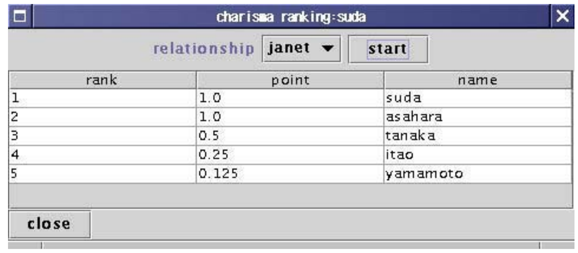

Figure 6. Screenshot of reputation ranking.

of communities who are related directly or indirectly beyond physical spaces and to expand the existing community or to form a new one as mentioned previously.

In a sense, relationships that constitute a community are mechanically created through chats. It may not be considered that a relationship between network peers represents a human relationship with the partner since the judgment of users is not contained in creating relationships. Meet@ compensates for this by using an attribute of relation strength in a relationship. This attribute is strengthened or weakened by explicit evaluations of the partner. A good evaluation strengthens it, and vice versa. Relation strength is used for judging whether a relationship is deleted or allowed to remain. Concretely, relationships with weak strength are deleted with a high probability at the time of creating a new relationship, and vice versa. This enables peers to be sure that their relationship is a human relationship with a partner that is reflected by the evaluation of each user. It also suppresses the increase in the number of relationships which each user holds. Thus, the community is reformed adaptively so that deserted users, who are not preferred at all, are removed from the community, which means that the deserted users are invisible from others since nobody has a relationship with them.

4.3. Ranking members of a community

A community is a web of relationship that represents a subjective reputation of the partner from the local viewpoint of each member about a keyword of the relation name. It is necessary to find reputable members throughout the community relevant to the keyword, including unseen members. Meet@ achieves this by using distributed pagerank, which calculates an objective reputation of each member from the global viewpoint of a community for ranking members of the community (see Figure 6). This reputation ranking is calculated based on the topology of the community, which in turn, is reflected by the evaluation of each member. Thus, the reputation ranking is considered to be appropriately reflected by the evaluation of each member.

The distributed calculation of pagerank is performed by each member, who knows only those with whom he has a

Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW'04)
0-7695-2050-2/04 $20.00 © 2004 **IEEE**

relationship, on the occasion to send and receive chat messages with each other asynchronously and constantly even under the dynamic topology of relationships.

5. Conclusion

This paper has proposed distributed pagerank as a distributed reputation model for open peer-to-peer networks. This model was inspired by the remark that pagerank has already satisfied the requirements of reputation models. Distributed pagerank needs no centralized calculation server and calculates the pagerank of each peer as its reputation in distribution, which is compatible with peer-to-peer networks. Distributed calculation of pagerank is performed asynchronously by each peer as it communicates with others. The asynchronous calculation accomplishes both to demand no extra message only for the calculation of pagerank and to calculate an accurate pagerank of each peer steadily even under the dynamic topology of relationships.

The result of the simulation has indicated that the calculated pagerank value of each peer converges at the original pagerank value under the static topology of relationships, which is presumable under a dynamic topology. An application of distributed pagerank has also been presented, which supports dynamic formation of communities with reputation ranking. These evaluations, however, have not yet been thoroughly described. This warrants future mathematical analysis to prove the convergence of distributed pagerank and an actual field test to clarify scalability and availability of the application using distributed pagerank. Nonetheless, it is notable that distributed pagerank has been newly suggested as a desirable reputation model for peer-to-peer networks on the basis of promising simulation results and a fully implemented application.

References

[1] K. Terada and T. Araragi. Calculating agent reliability in agent communication. *IEICE Technical Report*, 1999.

[2] R. Chen and W. Yeager. Poblano - a distributed trust model for peer-to-peer networks. *JXTA Security Project White Paper*, 2001.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Stanford Digital Libraries Working Paper*, 1998.

[4] Google search engine. http://www.google.com/

[5] K. Sankaralingam, S. Sethumadhavan, and J. C. Browne. Distributed pagerank for p2p systems. *Proc. of the IEEE HPDC-12*, 2003.

[6] T. Itao, T. Nakamura, M. Matsuo, T. Suda, and T. Aoyama. Service emergence based on relationship among self-organizing entities. *Proc. of the IEEE SAINT-2002*, 2002.

[7] T. Itao, T. Nakamura, M. Matsuo, S. Tanaka, T. Suda, and T. Aoyama. Relationship mechanism for dynamic and user preference-aware service creation. *IPSJ Journal*, 2003.

0-7695-2050-2/04 $20.00 © 2004 IEEE