

Optimization of routing strategies for data transfer in peer-to-peer networks

Atsushi Morioka and Akito Igarashi

Department of Applied Mathematics and Physics, Kyoto University, Kyoto JAPAN

igarashi@i.kyoto-u.ac.jp

Abstract. Since peer-to-peer file-sharing systems have become familiar recently, the information traffic in the networks is increasing. Therefore it causes various traffic problems in peer-to-peer networks. In this paper, we model some features of the peer-to-peer networks, and investigate the traffic problems. Peer-to-peer networks have two notable characters. One is that each peer frequently searches for a file and download it from a peer who has the requested file. To decide whether a peer has the requested file or not in modelling of the search and download process, we introduce file-parameter P_j , which expresses the amount of files stored in peer j . It is assumed that if P_j is large, peer j has many files and can meet other peers' requests with high probability. The other character is that peers leave and join into the network repeatedly. Many researchers address traffic problems of data transfer in computer communication networks. To our knowledge, however, no reports focus on those in peer-to-peer networks whose topology changes with time. For routing paths of data transfer, generally, the shortest paths are used in usual computer networks. In this paper, we introduce a new optimal routing strategy which uses weights of peers to avoid traffic congestion. We find that the new routing strategy is superior to the shortest path strategy in terms of congestion frequency in data transfer.

1. Introduction

Recently, studies on networks are extended to many fields such as human social networks, computer networks, neural networks and so on. In particular, complex networks, which have remarkable characters such as small world and the scale-free distribution of degrees, are attracting a great deal of attention [1,2,3]. Many studies have been addressed to information traffic congestion in the networks. Many users send data to other users through computer networks and traffic congestion happens. Efficient strategies for reducing the traffic congestion have been considered in recent years [4].

In this paper, we focus on peer-to-peer file-sharing networks [5,6,7] on computer networks. Since, in these days, applications for peer-to-peer networks, such as Napster, Gnutella, and Winny have been popularized, the amount of traffic in peer-to-peer networks is increasing yearly and becomes a major part of whole one in the Internet.

Therefore, avoidance of traffic congestion in peer-to-peer networks is an important problem. Although there are many papers that note the ways to reduce traffic congestion at the Internet, we recognize no papers which focus on peer-to-peer networks to our knowledge. Thus, focusing on characteristic aspects of peer-to-peer networks, we address traffic congestion on the networks.



2. Peer-to-peer networks

The structures of peer-to-peer networks are quite different from that of usual computer networks. There are main computers called "central servers" in the usual computer networks and all computers can be servers or clients.

2.1. Search for files and download of them

In the peer-to-peer networks each computer is called a "peer". A peer forwards a query of a file to its neighbors and the neighbors forward the query to their neighbors. This process is repeated until a specified threshold TTL (Time-To-Live) is reached. Here TTL is the threshold of the number of links where a search query passes. Queries are discarded if the number of the passed peers exceeds TTL. We assume that a peer can search files in the searchable domain in which all peers within a distance of TTL from the peer are included. A peer can search for a file and get data of the file from a "host peer", which has the requested file and exists in the searchable domain. Inversely, the peer can be a host peer for the other peers if it exists in their searchable domains. When a host peer sends data of a file, it is to be desired that the routing path is controlled for reduction of traffic congestion.

2.2. Leave and join in the peer-to-peer networks

The other character of the peer-to-peer networks is that peers leave and join in the networks by means of switching on or off the application which is needed for the peer-to-peer technique. Peers leave a peer-to-peer network for the reason that they have got files and need not connect to it any longer, and their PCs are gone down. On the other hand, peers join the network in order to get files one after another.

3. Model

We model a peer-to-peer file-sharing network, where peers leave and join in the network. The peer j has a file-parameter expresses the amount of stored files P_j . The distribution of P_j is given as [3] $\psi(p) = A \exp(-5p)$, where A is a positive constant, $0 \leq p \leq 1$, and P_j is normalized as the maximum value is set to be one. In the peer-to-peer networks, there are the peers who principally supply files to other peers. Empirically, these peers rarely leave the network. We call such peers "file suppliers", whereas peers who hardly supply files and quite often leave the network after having just joined are called "free riders". Therefore, we assume that the less files peers have, the more often they leave the network.

On the other hand, peers newly joining in the networks need information of peers who are already in the network in order to connect to the other peers. Since it is difficult to find information by themselves for the new peers, the administrator of the application for the peer-to-peer network gives information about peers. The administrator tends to select nodes which are in the network for a long time, because he or she hope that the network is held firm. Thus when peers join in the network, they often provide links to selected peers, who are file suppliers, namely have many files. To sum up these things, peers join in and leave the network in accordance with the next procedure.

- Some peers leave the network. The smaller P_j is, the more often peer j leaves.
- Due to the leaves, peers whose degree is 0 often appear in the network. Therefore in this time the respective peers add a link to a randomly selected peer whose degree is nonzero.
- The leaving peers join into the network again. The larger P_j is, the more often the rejoining peer connects a link to the peer j . The number of neighbors of the rejoining peer is set to be from 3 to 7 at random.

We obtain good agreement between theoretical and simulation results

4. Betweenness and routing strategy

First, a peer A searches for files which they want and forwards a query to its neighbour peers. This searching method is called "flooding". After the search, if the file which peer A wants exists in the

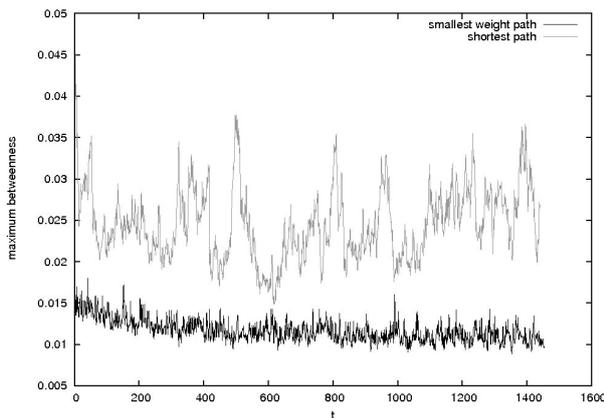
searchable domain, peer *A* downloads the file from peer *B* which has it in the domain. If a lot of data concentrate on a node, the node cannot transfer data to next node immediately, and it takes a lot of time to send files. Therefore, we need disperse transfer routing paths.

4.1. Betweenness

We consider that paths for all pairs of nodes in the network are defined according to a rule, for example, choosing the shortest path and so on. Betweenness of a peer is defined as a measure for concentration of optimal paths at the peer.

4.2. Routing Strategy

First, we introduce the shortest path strategy. Transport routing strategies used on computer networks have been solely determined based on the shortest paths. However in the case of shortest path strategy, traffic concentrates on peers with a great many degrees, and one of the peers have the big maximum betweenness B_{max} . In finding optimal (smallest) B_{max} , we need to disperse routing paths. For the



purpose, we employ the smallest weight path algorithm, that a path with the smallest sum of weights assigned to peers along the path is chosen. We show B_{max} at both the shortest path strategy and new strategy which uses the smallest weight path algorithm.

Fig1. $B_{max}(t)$ is plotted as a function of t . The grey line represents the results with the shortest path algorithm and the black line those with the smallest weight path algorithm where the number of peer $N=2000$ and $TTL=4$. The number of peers which leaves and joins at one step is 50

A. Shortest path algorithm

1. Leaves and joins of peers are performed.
2. Select efficient paths, where the total summation of the number of the links in each path is minimum between each pair of peers in the searchable domain.
3. Compute the betweenness of every peer for the efficient paths.
4. $t=t+1$ and repeat 1 to 3.

B. Smallest weight path algorithm

1. Assign the initial weight $W_j(0) = k_j^{0.7}$ to every node, where k_j denotes the degree of peer j
2. Leaves and joins of peers are performed
3. Select efficient paths, where the total summation of the weights of the peers in each path is minimum between each pair of peers in the searchable domain.
4. Compute the betweenness of every peer for the efficient paths.
5. Update the weight of nodes $W_j(t)$ at t which satisfies $\frac{B_j(t)}{B_{max}(t)} > 0.9$ as follows:

$$W_j(t+1) = W_j(t) + \left(\frac{B_j(t)}{B_{max}(t)} \right)^{100} \text{ where } B_j(t) \text{ is the betweenness of peer } j \text{ at } t \text{ and } B_{max}(t) = \max_j B_j(t)$$

6. $t=t+1$ and repeat 2 to 5

4.3. Calculation $B_{max}(t)$

In Fig.1, we show $B_{max}(t)$ for both the shortest path algorithm and smallest weight path algorithm. We find that smallest weight path algorithm is superior to shortest path algorithm because B_{max} of the algorithm is stably lower than one of the shortest path algorithm.

5. Data Transfer

5.1. Data transfer model

1. *Select randomly a peer who wants a file.*
2. *The peer A selected in 1. searches for peers who have the requested file in its searchable domain by flooding.*
3. *A investigates the number of files P which each peer in searchable domain has, and the larger P a peer has, the more often the peer (peer B) holds the requested file and peer B is selected as a peer who sends its data to peer A selected in 1.*
4. *Peer B starts data transfer to peer A. At one step, a peer sends data to a neighbour.*

We perform these four procedures with some peers simultaneously. To perform data transfer with many peers lead to traffic congestion. In next section we define congestion in data transfer.

5.2. Congestion

Through data transfer, we assume that all peers can send two different data to its neighbour at once. If a peer receive three different data at the same step, at next step the peer can send only two data, and consequently it leads one data to stop at the peer . We define this phenomenon as congestion.

<i>The number of peers requiring files</i>	<i>Shortest Path</i>	<i>Smallest Weight Path</i>
	<i>congestion</i>	<i>Congestion</i>
<i>300</i>	<i>36</i>	<i>12</i>
<i>400</i>	<i>109</i>	<i>59</i>
<i>500</i>	<i>149</i>	<i>89</i>
<i>600</i>	<i>233</i>	<i>146</i>

Table.1 The number of leaving peers is 50. The value is the number of times traffic congestion occurs.

5.3. Simulation

When leaving peers are selected in the path of data transfer, the transfer path changes in order to send data to the peer who wants a file as much as possible.

From Table 1, we find that the number of times traffic congestion occurs is smaller in the case of

the smallest weight path than in the case of the shortest path.

References

- [1] D.J. Watts, S.H. Strogatz, "Collective dynamics of small-world networks", Nature Vol.393 (1998) pp.440-442.
- [2] A.L. Barabasi, R. Albert, "Emergence of scaling in random networks", Science Vol.286 (1999) pp.509-512.
- [3] M.E.J. Newman, "The structure and function of complex networks", SIAM Review Vol.45 (2003) pp.167-256..
- [4] H. Kawamoto and A. Igarashi, "Efficient packet routing strategy in complex networks", Physica A vol.391 (2011) pp.895-904.
- [5] M. Ripeanu, I. Foster and A. Iamnitchi, " Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design", IEEE Internet Computing Journal vol.6 ,no.1 pp.50-57 (2002)..
- [6] D. Stutzbath and R. Rejaje, "Charactering today's gnutella topology", Technical Report, CIS-TR-04-02, Department of Computer Science, University of Oregon (2004).
- [7] T. Maruyama, "Virus spreading model of peer-to-peer networks", Master's thesis, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University (2009)