This paper was presented as part of the main technical program at IEEE INFOCOM 2011

A Model Approach to Estimate Peer-to-Peer Traffic Matrices

Ke Xu
Department of Computer Science
Tsinghua University
Beijing, P.R.China
xuke@csnet1.cs.tsinghua.edu.cn

Meng Shen
Department of Computer Science
Tsinghua University
Beijing, P.R.China
shenmeng@csnet1.cs.tsinghua.edu.cn

Mingjiang Ye*
Research and Development Center
Yahoo
Beijing, P.R.China
mye@yahoo-inc.com

***Abstract*—Peer-to-Peer (P2P) applications have become increasingly popular in recent few years, which bring new challenges to network management and traffic engineering (TE). As basic input information, P2P traffic matrices are of significant importance for TE. Due to excessively high cost of direct measurement, a lot of studies aim at modeling and estimating general traffic matrices, but few focus on P2P traffic matrices. In this paper, we proposed a model to estimate P2P traffic matrices in networks. Important factors are considered, including the number of peers, the localization ratio of P2P traffic, and the distances among different networks. Here distance can be hop counts or geographic distance accordingly. To validate our model, we have evaluated the performance using both real P2P live steaming traces and file sharing application traces. Evaluation results show that the proposed model outperforms the other two typical models for general traffic matrices estimation, in terms of estimate errors. To the best of our knowledge, this is the first research on P2P traffic matrices estimation. P2P traffic matrices, derived from the model, can be applied to P2P traffic optimization and other TE fields.**

***Index Terms*—Traffic matrix, Peer-to-Peer (P2P), Traffic engineering**

I. Introduction

Knowing how traffic flows through the network is not trivial work to network operators in network design and management,including traffic engineering, failure recovery, bandwidth provision, etc. The distribution of network traffic can be presented by a traffic matrix (TM) which gives traffic volume between each origin and destination (OD) pair in the network.

Estimation approaches based on partial network information are well accepted to generate traffic matrices because of excessively high cost of direct online measurement. The estimation problem can be briefly described as follows. Let y be the vector of link counts, x be the traffic matrix reorganized as a column vector. The routing matrix is denoted by A, where A_{ij} is equal to 1 if route i belongs to OD pair j or 0 otherwise. Then the relationship among link counts, traffic matrix and routing matrix can be expressed as $y = Ax$. We can obtain the link counts y and routing matrix A through SNMP measurements and IGP link weights together with network topology information, respectively. However, The computation of traffic matrix x from the above equation is not straightforward, since the number of OD pairs is far more than that of the link counts. The matrix A is thus less than full rank, making the fundamental problem an ill-posed system.

Researchers in recent years have proposed a variety of methods and models to make estimation processes more convenient and precise, which are well summarized in [4]. Few of these studies, however, are concerned with estimation on Peer-to-Peer (P2P) traffic matrices.

Peer-to-peer systems have gained tremendous popularity in the past few years. Numerous studies present that traffic generated by P2P systems and applications accounts for a major fraction of the Internet traffic [10]. The large volume of P2P traffic significantly increases the load on the Internet, making networks more vulnerable to congestion and failures, and hence brings new challenges to efficiency and fairness of networks. There has long been a desire for ISPs to improve overlay routing schemes in a friendlier means for both users and ISPs.

The model proposed in the paper is based on observation of important features in real P2P systems. Multi-connections are concurrently established for data exchange and transmission between a host and a subclass of all its neighbors, which are called *peers* in P2P systems. The number of concurrent connections and the choice of peers are decided by peer selection policies and thus differ among different P2P systems and applications. It implies that the volume of data transmission will be greatly affected by the number of peers in networks and the distance between them. Here, distance can be described via overlay virtual distance, AS hops or even geographic Euclid distance. Moreover, P2P traffic localization is another important issue.

To capture these crucial features, we proposed a model to estimate P2P traffic matrices. As mentioned in [8], the model can be used to generate or characterize traffic matrices for a given network topology or parameters with physical meanings. It takes the following physically meaningful factors into consideration. The number of peers is the first to be considered. Intuitively, networks with more peers should have larger volumes of P2P traffic. Another factor is the traffic localization ratio, which covers the internally exchanged

This work has been supported in part by NSFC Project(60970104), 973 Project of China (2009CB320501), 863 Project of China (2008AA01A326) and Program for New Century Excellent Talents in University.

* This work was done partially by Ye when he was in Dept. of CS, Tsinghua University

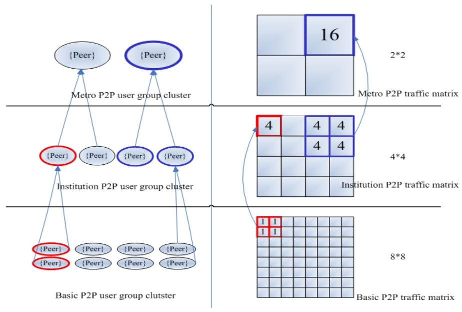


Fig. 1. P2P traffic matrices aggregation process

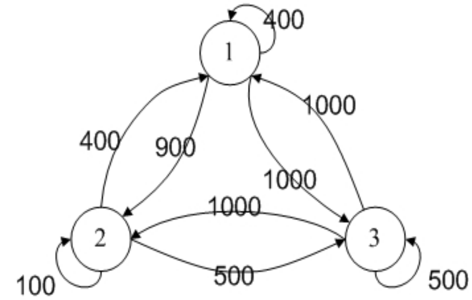


Fig. 2. A simple example of P2P traffic matrix

portion of P2P traffic. And the last but not least one is the distance between different networks. As is mentioned above, distance can precisely reflect the peer selection strategy of the concerned system.

Using real P2P traffic traces of both P2P live streaming applications and P2P file sharing systems, we evaluate the performance of our model through comparison with another two typical models proposed for general traffic matrices estimation, namely the gravity model [1] and the independent connection model [8]. Evaluation results show that the newly proposed P2P model outperforms the other two models in terms of estimate errors.

The rest of this paper is organized as follows. In Section II, we illustrate the methodology as a guideline to develop the model, following which the model to estimate P2P traffic matrices are formally presented in Section III. We have evaluated our model by real P2P traffic traces in Section IV and discussed other issues of our model in Section V. After briefly looking back to the related work in Section VI, we conclude this paper in Section VII.

II. Terminology and Methodology

In this section we will present several important notations and illustrate the methodology as a guideline to develop a precise model.

Generally, an element X_{ij} in traffic matrix X represents traffic volume flowing from the original node i to the destination node j during a certain time interval. In previous works on general traffic matrices, since researchers mainly focused on the traffic exchanged between links or routers, a *node* is conventionally referred to as a router. In our mode, however, we will focus on the application layer traffic which reflects application or user behavior. The *node* in our model thus represents a P2P user group comprised of several P2P hosts.

We assume that the network has N P2P users altogether, denoted by $H(< h_1, h_2, \cdots, h_N >)$. Each P2P user h_i can form a basic P2P user group with the lowest granularity.

The basic P2P user groups can be further aggregated into high-level P2P user groups as shown in Fig. 1. We can merge the basic P2P user groups within the same institution network into a new institution P2P user group. Similarly, the institution user groups can also be clustered into metro user groups. Therefore, a new user group with higher granularity can always be generated through gathering several P2P user groups at relatively low level, until all peers are involved in the same group.

We use H_i^k to represent the ith P2P user group, where k is the aggregation level. And H^{k+1} thus is the cluster of $H_i^k(t)$, which can be represented by Eq. 1.

$$H^{k+1} = \{H_i^k \mid i \in \{1, \cdots, |H^{k+1}|\}\} \tag{1}$$

According to the above definition, H^1 is the basic P2P user group cluster, while H^2 is the aggregation of $H^1(t)$.

While investigating P2P traffic between different institutions, we are interested in the P2P traffic exchanged among institution P2P user groups and try to figure out the institution P2P traffic matrix. Similar to user groups, P2P traffic matrices with higher granularity level can be calculated by aggregating relatively low level P2P traffic matrices.

The P2P traffic matrix X^k represents the P2P traffic exchanged between different P2P user groups H_i^k during a certain time interval. The element X_{ij}^k of a P2P traffic matrix can be directly inferred from elements of the P2P traffic matrix with a lower level, which is illustrated by Eq. 2.

$$X_{ij}^k = \sum_{\forall s, H_s^{k-1} \subset H_i^k} \sum_{\forall t, H_t^{k-1} \subset H_j^k} X_{st}^{k-1} \tag{2}$$

I_i^k and E_i^k are total P2P traffic volumes of H_i^k sent to and received from $H_j^k (j \neq i)$, respectively.

$$I_i^k = \sum_{j=1}^{j=n} X_{ij}^k (j \neq i) \quad E_i^k = \sum_{i=1}^{i=n} X_{ij}^k (i \neq j) \tag{3}$$

Assume that C_i^k is the amount of traffic exchanged within H_i^k, the total uploading and downloading volume of P2P traffic in H_i^k, denoted respectively by U_i^k and D_i^k, can be derived from Eq. 4.

$$U_i^k = I_i^k + C_i^k \quad D_i^k = E_i^k + C_i^k \tag{4}$$

Let $|H_i^k|$ denote the number of peers in H_i^k, and the portion of peers in H_i^k to the total peers is $\mu_i^k = |H_i^k|/N$. The network distance between H_i^k and H_j^k is d_{ij}^k, which can be measured by hop counts, RTT, or geographic distance between two nodes.

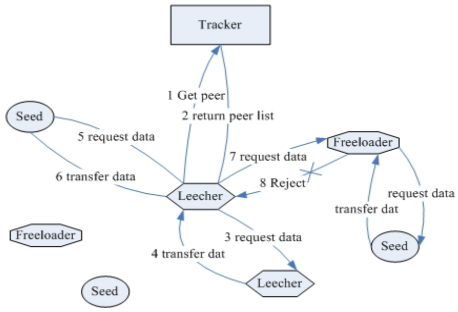


Fig. 3. The download process of BitTorrent

For simplicity, we present a simple topology with three nodes representing three user groups in Fig 2. Assume that the aggregation level denoted by k equals to 4, and the time interval t is omitted here. The numbers marked beside arrows denote the volume of P2P traffic along the directions of arrows, whose unit is omitted. Take Node 1 for example, $X^4_{11} = 400$, $X^4_{12} = 900$, $X^4_{21} = 400$. The total uploading and downloading volume of P2P traffic for Node 1, denoted by U^4_1 and D^4_1, are 2300 and 1800, respectively.

III. The Model for P2P Traffic Matrices

In this section we first deeply analyze characteristics of BitTorrent system [13], based on which the model to estimate P2P traffic matrices is proposed. Then processes of parameters learning and P2P traffic matrices generation are presented.

A. Characteristics in P2P Systems

In this subsection, we will have a deep insight into P2P systems as the base of modeling P2P traffic. Here we take BitTorrent [13], a typical and popular P2P file sharing application, as an example to analyze characteristics in P2P systems.

BitTorrent is a traditional P2P file sharing application. The basic process is as follows. A big file will be divided into small data pieces and BitTorrent users will download data pieces in need from other peers. Like other P2P applications, each BitTorrent user will only have a few other BitTorrent users as neighbors which improve the overall scalability.

Recent studies found peer behaviors are quite different [14]. Some peers upload a lot of data but seldom download, while some peers download a lot of data but seldom upload. So we can roughly classify peers into the following three categories:

- **Seeds**: peers that upload a lot of data but never download. In BitTorrent, seeds do not have bias against uploading.
- **Free-riders**: peers that download a lot of data but seldom upload. Free-riders are more likely to reject the data request from other P2P peers.
- **Leechers**: peers that not only download but also upload data. In BitTorrent, leechers prefer uploading to peers who have uploaded more data to them before.

The download process is shown in Fig. 3, which can be divided into three phases: peer requirement, data request and data transmission.

In peer requirement phase, a peer newly joining in the system will request a list containing the information of other peers from one or several central servers which are called *trackers*. The list only contains partial peers in the system. In the mainstream implementation of trackers, the peers in the list are selected randomly without any bias. But recently, a lot of researchers focused on improving the phases, such as P4P, Oracle [15] [16]. Their common idea is that a host prefers to select their neighboring peers to improve the performance. The network distance is either measured by peers themselves or provided by ISP-operated services. In the case, P2P peer selection is related to the network distance.

In the next phase, a peer will send data requests to other peers on the list. According to the default setting in BitTorrent, a host could only concurrently upload to at most four peers, so the data requests might be rejected. Leechers will prefer responding to the data request from peers who have uploaded to them, while free-riders will reject the majority of the data requests they receive.

Connections are set up between a host and each of its peers who have accepted data requests, and then the data transmission phase will begin.

B. Modeling Basic P2P Traffic Matrices

We refer to the traffic exchanged between each pairs of peers as *basic P2P traffic matrices*. To simplify the analysis, the dynamic of peers in P2P systems are not considered. We built up a probability model to capture basic P2P traffic matrices as the basis of modeling high-level P2P traffic matrices.

Hereafter we use G_s, G_f and G_l to respectively denote the cluster of seeds, free-riders and leechers.

Considering two peers h_i and h_j, the process of h_i sending data to h_j can be broken down as follows. Firstly, h_j gets the peer list from trackers, in which the probability of containing h_i is denoted by P^s_{ji}. Assume that the data request rate from h_j to h_i is T_{ji}, and the probability of h_i responding the data request from h_j is P^r_{ij}. Finally, h_i begins to transfer data to h_j with the flow throughput B_{ij}. So we know that the traffic from h_i to h_j is:

$$X_{ij} = P^s_{ji} T_{ji} P^r_{ij} B_{ij} \tag{5}$$

Now we will go through each parameters in Eq. 5. For h_i who is not seed, the probability of not getting the specific h_j from tracker in the i-th place of the list is $1 - 1/(N - i) * d^s_{ji}$, where N is the total number of P2P users in the system and d_{ji} is the network distance between h_i and h_j. Peer selection factor s is nonnegative, if it equals to 0, trackers will return a list of peers randomly; otherwise, trackers will take the network distance into consideration. Assuming h_i retrieves L_i peers from the tracker, the probability of not getting h_j, denoted by $\overline{P^s_{ij}}$, can be derived from Eq. 6.

$$\overline{P^s_{ij}} = \prod_{\psi=1}^{L_i} (1 - \frac{1}{(N-\psi)(d_{ij})^s}) \tag{6}$$

Since $N \gg L_i$ always holds in P2P systems, the probability of getting h_j is:

$$P_{ij}^s = 1 - \overline{P_{ij}^s} \approx 1 - (1 - \frac{1}{N(d_{ji})^s})^{L_i} \approx 1 - (1 - \frac{L_i}{N(d_{ji})^s}) = \frac{L_i}{N(d_{ji})^s} \quad (7)$$

Seeds will not download data, so they will not retrieve the list. Therefore, for both seed and non-seed peers, we have:

$$P_{ij}^s = \begin{cases} 0, & h_i \in G_s \\ \frac{L_i}{N(d_{ji})^s}, & h_i \notin G_s \end{cases} \quad (8)$$

Seeds will not request data, so their data request rates always equal to 0. For h_i who is not seed, assume that h_i sending out R_i request each time period and it had M_i members. Ignoring the differences of the downloaded data of each peer, we regard data request from h_i to h_j as R_i/M_i. And T_{ij} is thus presented as follows:

$$T_{ij} = \begin{cases} 0, & h_i \in G_s \\ \frac{R_i}{M_i}, & h_i \notin G_s \end{cases} \quad (9)$$

Since free-riders will reject the entire data request, P_{ij}^r is 0 when h_i belongs to free-riders. When the service capacity S_i is not less than demand D_i, P_{ij}^r is 1; Otherwise, P_{ij}^r is S_i/D_i. Leechers will prefer uploading to peers who have uploaded to them before. When S_i is not less than demand D_i, P_{ij}^r is also 1. But when S_i is insufficient, the P_{ij}^r is related to P_{ji}^r. We can get:

$$P_{ij}^r = \begin{cases} 0, h_i \in G_f \\ min(1, \frac{S_i}{D_i}), h_i \in G_s \\ \left.\begin{matrix} 1, S_i \geq D_i \\ 0, S_i < D_i \end{matrix}\right\}, h_i \in G_l, h_j \in G_f \\ \left.\begin{matrix} 1, & S_i \geq D_i \\ \frac{p_{ji}^r}{\sum p_{ji}^r} \frac{S_i}{D_i}, & S_i < D_i \end{matrix}\right\}, h_i \in G_l, h_j \in G_l \end{cases} \quad (10)$$

When two peers begin to transfer data, the data transmission rate is not relevant to the type of peers, but only to the flow throughput. From the classical TCP performance model [21], we can obtain:

$$TCP_{BW} = \frac{C * MSS}{RTT\sqrt{p}} \quad (11)$$

where C is the number of TCP ACK packets, MSS is the maximal segment size, RTT is the round trip time, and p is the packet drop probability. Since they can all be considered as the metrics of network distance, B_{ij} is in proportion to $1/d_{ij}^s$. And s is the transfer performance factor of network distance. Assume the bottleneck is the upload capacity rather than the download capacity, which is a common assumption in modeling the performance of P2P system [14]. Then the upload capacity of h_i is allocated to different peers according to the weight of $1/d_{ij}^s$.

Therefore, we can deduce the following expression:

$$B_{ij} \propto 1/d_{ij}^s \Rightarrow B_{ij} = \frac{1/(d_{ij})^s}{\sum_j 1/(d_{ij})^s} U_i \quad (12)$$

From the previous analysis, we can get the basic P2P traffic between different types of peers as shown in TABLE I. Seeds have no incoming traffic, while free-riders have no outgoing traffic.

TABLE I
BASIC P2P TRAFFIC MATRIX

Type	G_s	G_f	G_l
G_s	0	X_{sf}	X_{sl}
G_f	0	0	0
G_l	0	X_{lf}	X_{ll}

Data exchanged thus can be derived from Eq. 13:

$$\begin{aligned} X_{ij}^{sf} &= X_{ij}^{sl} = \frac{R_j L_j}{M_j N} \frac{U_i}{(d_{ij})^s} * min(1, \frac{S_j}{D_j}) * \frac{1/d_{ij}^t}{\sum_j 1/(d_{ij})^s} \\ X_{ij}^{lf} &= \frac{R_j L_j}{M_j N} \frac{U_i}{(d_{ij})^s} * \left\{\begin{matrix} 1, S_i \geq D_i \\ 0, S_i < D_i \end{matrix}\right\} \frac{1/(d_{ij})^s}{\sum_j 1/(d_{ij})^s} \\ X_{ij}^{ll} &= \frac{R_j L_j}{M_j N} \frac{U_i}{(d_{ij})^s} * \left\{\begin{matrix} 1, & S_i \geq D_i \\ \frac{p_{ji}^r}{\sum p_{ji}^r} \frac{S_i}{D_i}, & S_i < D_i \end{matrix}\right\} \frac{1/(d_{ij})^s}{\sum_j 1/(d_{ij})^s} \end{aligned} \quad (13)$$

The model of basic P2P traffic matrices has assumed that the whole system considered is in stable status. The final traffic matrix is a probability matrix. It is thus hard to get accurate P2P traffic matrices by using the model directly. But after aggregation, P2P user groups and P2P traffic exchanged between different user groups will have statistical characteristics. The model of basic P2P traffic matrix is the basis of our high-level model.

C. Modeling High-level P2P Traffic Matrices

In Section III.A, we have illustrated that the k-level P2P traffic matrix can be inferred from the $(k-1)$-level P2P traffic matrix via Eq. 2. Therefore, through aggregation, we can achieve the k-level P2P traffic matrix from the basic level P2P traffic matrix.

$$X_{ij}^k = \sum_{h_m \in H_i^k} \sum_{h_n \in H_j^k} X_{mn}^1 \quad (14)$$

where X_{mn}^1 denotes the basic level P2P traffic matrix derived from Eq. 5.

Now we are in the position of exploring the statistical characteristics in high-level P2P traffic matrix so as to obtain the model for estimation.

Considering P2P user groups H_i^k and H_j^k with population of peers denoted by $|H_i^k|$ and $|H_j^k|$, respectively. Then P_{ji}^s is the probability of containing peers belonging to H_i^k on the peer lists of users in H_j^k getting from tracker severs, it thus should be proportional to peer ratios of these two groups, μ_i and μ_j, and inverse proportional to the network distance d_{ij}.

$$P_{ji}^s \propto \frac{|H_i^k||H_j^k|}{(d_{ij})^s} \propto \frac{\mu_i^k \mu_j^k}{(d_{ij})^s} \quad (15)$$

For T_{ji}, the request rate of H_j is proportional to the total download capacity of H_j, and the response probability P_{ij}^r is proportional to service capacity of H_i, as is illustrated in Eq. 16:

$$T_{ji} \propto D_i, \quad P_{ij}^r \propto U_i \quad (16)$$

For H_i and H_j, the flow throughput between them is inverse proportional to the network distance.

$$B_{ij} \propto \frac{1}{(d_{ij})^s} \tag{17}$$

Then we derive the model to estimate high-level P2P traffic matrix, as is shown below:

$$X_{ij} = \frac{K\mu_i\mu_j}{(d_{ij})^s} U_i D_j \tag{18}$$

where K is a scale factor and s is the distance factor of peer selection and flow throughput. The larger the value of s is, the more important of the distance factor will be in peer selection process and the more impact of network distance will have on transmission performance.

Considering the time series, the model can be rewritten as follows:

$$X_{ij}(t) = \frac{K\mu_i(t)\mu_j(t)}{(d_{ij})^s} U_i(t) D_j(t) \tag{19}$$

Compared with the simple gravity model in [1], the model uses $U_i(t)$ and $D_j(t)$ as the repulsive factors instead of $I_i(t)$ and $E_i(t)$.

$$X_{ij} = \frac{I_i E_j}{\sum_k E_k} \tag{20}$$

As shown in Eq. 20, X_{ij} and X_{ji} are independent in the simple gravity model. The assumption is valid in real traffic [8]. P2P traffic in the forward and reverse directions has strong dependence because peers in the two networks exchange data in both directions of the connection.

D. Parameters Learning

The parameters in our model have their physical meanings, such as the distance matrix d_{ij} representing distance among networks in terms of hop counts or geographic distance. In order to obtain an accurate model to estimate P2P traffic matrix, the learning process is necessary to adjust parameters according to real historical traffic traces.

By using the number of peers and the average upload and download rates from the measurement, $U_i(t)$ and $D_j(t)$ can be estimated. But $U_i(t)$ and $D_j(t)$ can be also inferred by $I_i(t)$ and $E_j(t)$, which can be measured at the edge routers of the nodes.

$$\begin{aligned} U_i(t) =& \frac{I_i(t)}{I_i(t)/U_i(t)} = \frac{I_i(t)}{1 - C_i(t)/U_i(t)} = \frac{I_i(t)}{1-\alpha_i(t)} \\ D_i(t) =& \frac{E_i(t)}{E_i(t)/D_i(t)} = \frac{E_i(t)}{1 - C_i(t)/D_i(t)} = \frac{E_i(t)}{1-\beta_i(t)} \end{aligned} \tag{21}$$

$\alpha_i(t)$ and $\beta_i(t)$ are the traffic localization factor, where

$$\alpha_i(t) = \frac{C_i(t)}{U_i(t)}, \quad \beta_i(t) = \frac{C_i(t)}{D_i(t)} \tag{22}$$

$\alpha_i(t)$ is thus the P2P traffic fraction exchanged internally out of the total upload of P2P traffic, while $\beta_i(t)$ is the P2P traffic fraction exchanged internally out of the total downloading P2P traffic.

From Eq. 19 $\sim$ 22, $X_{ij}(t)$ can be rewritten as follows:

$$X_{ij}(t) = \frac{K\mu_i(t)\mu_j(t)}{(d_{ij})^s} \frac{I_i(t)}{1-\alpha_i(t)} \frac{E_j(t)}{1-\beta_j(t)} \tag{23}$$

Simple algebraic manipulation gives us:

$$\begin{aligned} \alpha_i(t) =& \frac{\beta_i(t)}{\frac{I_i(t)}{E_i(t)} + (1 - \frac{I_i(t)}{E_i(t)})\beta_i(t)} \\ \beta_i(t) =& \frac{\alpha_i(t)}{\frac{E_i(t)}{I_i(t)} + (1 - \frac{E_i(t)}{I_i(t)})\alpha_i(t)} \end{aligned} \tag{24}$$

From Eq. 23 and 24, we can get

$$X_{ij}(t) = \frac{K\mu_i(t)\mu_j(t)}{(d_{ij})^s} \frac{(1 + \frac{I_i(t)}{E_j(t)})\alpha_j(t) I_i(t) E_j(t)}{(1-\alpha_i(t))(1-\alpha_j(t))} \tag{25}$$

Then parameters learning process could be formed as an optimization problem, as is shown below, with the minimization of estimate errors as objective function.

$$\begin{array}{lll} \text{minimize} & RelL2(X_{ij}(t)) & \\ \text{subject to} & 0 \le \mu_i(t) \le 1, & \forall i \\ & 0 \le \alpha_i(t) \le 1, & \forall i \\ & d_{ij} = d_{ji}, & \forall i,j \\ & \sum_i \mu_i(t) = 1 & \end{array} \tag{26}$$

Motivated by [4], here we introduce relative L2 norm, which is shown in Eq. 27, as the metric of accuracy. $\hat{X}_{ij}(t)$ is the estimated value of the elements of the P2P traffic matrix by our model, while $X_{ij}(t)$ is the true value. The lower the RelL2 is, the more accurate the model is.

$$RelL2(t) = \frac{\sqrt{\sum_{i=1}^N \sum_{j=1}^N (X_{ij}(t) - \hat{X}_{ij}(t))^2}}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N X_{ij}(t)^2}} \tag{27}$$

We derive the values of $\mu_i(t)$, $\alpha_i(t)$, and d_{ij} from solutions of the above nonlinear program using optimization toolbox provided in Matlab.

E. P2P Traffic Matrices Estimation

By applying the parameters in Eq. 25, the P2P traffic matrix in the time interval t can be estimated. Assume that only the Ingress and Egress P2P traffic volume is available, we can also calculate the P2P traffic matrix. From Eq. 3, we can get:

$$\begin{aligned} \frac{k\mu_i(t)}{1-\alpha_i(t)} \sum_{j=1,j\neq i}^{n} \frac{\mu_j(t)E_j(t)}{(1-\beta_j(t))(d_{ij})^s} = 1, \forall i \\ \frac{k\mu_i(t)}{1-\beta_i(t)} \sum_{j=1,j\neq i}^{n} \frac{\mu_j(t)I_j(t)}{(1-\alpha_j(t))(d_{ij})^s} = 1, \forall i \end{aligned} \tag{28}$$

Setting $\upsilon_i(t) = \frac{k\mu_i(t)}{1-\alpha_i(t)}$ and $\nu_i(t) = \frac{k\mu_i(t)}{1-\beta_i(t)}$, we get the following relationships.

$$\begin{aligned} \upsilon_i(t) \sum_{j=1,j\neq i}^{n} \frac{\nu_j(t)E_j(t)}{(d_{ij})^s} = 1, \forall i \\ \nu_i(t) \sum_{j=1,j\neq i}^{n} \frac{\upsilon_j(t)I_j(t)}{(d_{ji})^s} = 1, \forall i \\ X_{ij}(t) = \frac{\upsilon_i(t)\nu_j(t)I_i(t)E_j(t)}{(d_{ij})^s} \end{aligned} \tag{29}$$

In accordance with 29, we can calculate $\upsilon_i(t)$ and $\nu_i(t)$ using Algorithm 1, with ingress P2P traffic $I(t) = [I_i(t)]$, egress P2P traffic $E(t) = [E_i(t)]$ and distance matrix d_{ij} in each time period. And then, P2P traffic matrices can be calculated.

Here we set the error threshold $\theta_{threshold}$ to be 10^{-5} based on experimental observations. With additional P2P traffic in the networks, such as P2P traffic in links, the estimated result of our model can be further used as the input of other tomographic methods. A more accurate input of the tomographic methods can lead to a more accurate result.

This approach has the following advantages for P2P traffic matrix generation. First, the parameters have physical meanings in real network. For example, one can adjust the distance factor of peer selection s and the traffic localization factor $\alpha_i(t)$ to analyze the distance impact in different P2P applications and networks. Besides, different distributions of peers in the network can be generated to investigate the impact of some popular P2P networks. Second, our model needs a few inputs: $nt + n + 2$ for a network of size n over t time steps.

Algorithm 1 *Iterative Algorithm*

```
Input: I,E,d_ij
Output: υ,ν
ν = [0.1, ..., 0.1] // an initialized value as the seed
while true do
    for i = 1 to N do
        υ_i = 1/ Σ_{j=1}^{N} ν_j E_j / (d_ij)^s (j ≠ i)
        ν' = ν
    end for
    for i = 1 to N do
        ν_i = 1/ Σ_{j=1}^{N} υ_j I_j / (d_ji)^s (j ≠ i)
    end for
    if ‖ ν − ν' ‖< θ_threshold then
        break;
    end if
end while
```

IV. Performance Analysis

In this section, we will evaluate the performance of our model in estimating Peer-to-Peer traffic matrix. As is illustrated above, estimation accuracy will partially depend on the choice of parameters. We thus firstly investigate the influence of parameters on P2P model performance, and then make a comparison among gravity model, independent connection model and our P2P model on estimation errors.

A. Evaluation Setup

Although public traces of general traffic matrix are available online, those of P2P traffic are not found yet. To evaluate the performance of our model, we collect datasets containing both P2P live streaming application traces and P2P file sharing application traces.

The former traces are collected form PPLive [12], a popular P2P live streaming application in China. The traces are comprised of traffic volume of live streaming data exchanged in six different ISPs in China: China Telecom, China Netcom, China Mobile, Cernet, China TieTong and the others (users beside the first 5 ISPs). And in the following evaluations, we use *pplive* dataset to indicate the traffic traces above.

And the latter traces are derived from running BT-like software on PlanetLab TestBed [19], which involves data exchange logs of 289 valid hosts all around the world. Then we get the geographical information of each host via mapping host IP to its corresponding latitude and longitude [20]. Based on the geographical coordinate, we derive a traffic matrix data set called *planet* dataset hereafter.

From the above description, we need to highlight that the two datasets derived are of different granulates. The former one is an ISP-level traffic matrix, while the latter one is at the host level. Therefore, if our model performs well using both of the above two datasets, we may deduce that it is an appropriate model to estimate P2P traffic matrices.

B. Evaluation of Parameters Influence

P2P model performance mainly depends on two categories of parameters, namely the distance matrix d_{ij} and distance factor s, respectively. In this subsection, we will use the above two datasets to investigate the impact of distance matrix d_{ij} and distance factor s on estimation accuracy.

A traffic matrix estimate yields an estimation value per OD flow, denoted by $\hat{X}_{ij}(t)$, for each time interval t. We can derive a set of error metrics across both time and space using the $X_{ij}(t)$ and $\hat{X}_{ij}(t)$. The two datasets mentioned above, however, both collect data within one time interval and do not contain a time series of flow estimates. Here t is thus set to be a fixed number, say 1, and we can observe the estimate errors across all OD flows for that fixed t. In the following evaluations, we use different ways to view and summarize these errors aiming at properly exploring the influence of parameters.

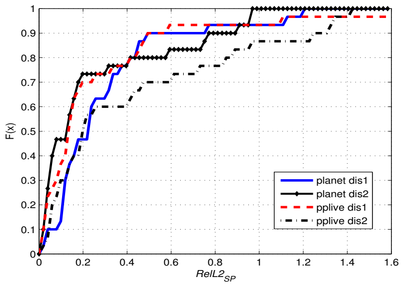
1) Distance matrix: In order to evaluate distance matrix d_{ij} using two datasets, for each dataset we separately derive two variations of our P2P model with two different sets of distance parameters.

For pplive dataset, in the first variation, distance elements between each ISP pair are all set to be 1, which is named pplive-1. The second variation is the pplive-2, where the distance elements between each ISP pair are all set to be 1, except that between China Telecom and China Netcom which equals to 2. The larger distance value between the above two ISPs is reasonable as bottlenecks between China Telecom and China Netcom observed in [11].

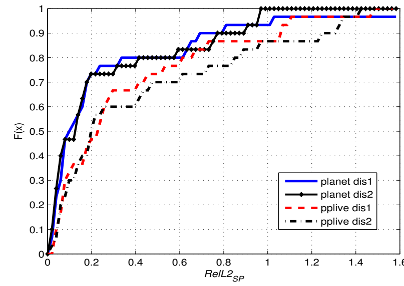
For planet dataset, the first variation is named planet-1. Similar to pplive-1, its distance elements are all set to be 1. And as the second variation, planet-2 sets distance elements between each OD pair according to their geographic distance.

Motivated by [4], we use the spatial errors as a metric of estimation accuracy, which is given as follows

$$RelL2_{SP}(i,j) = \frac{\sqrt{\sum_{t=1}^{T}(X_{ij}(t) - \hat{X}_{ij}(t))^2}}{\sqrt{\sum_{t=1}^{T} X_{ij}(t)^2}} \qquad (30)$$



(a) Distance factor s is 1



(b) Distance factor s is 0.3

Fig. 4. Relative L2 norm error in space

For each flow we compute the relative L2 norm thus deriving an error metric per flow. The flow error distribution with different values of distance factors s is given in Fig. 4 (a) and (b). We can see different distance factor s in both cases, and no matter which dataset is used for evaluation, the variation with distance matrix reflecting practical meanings between OD pairs always outperforms the one with all distance matrix elements set to 1 in terms of spatial relative L2 norm error. Appropriate distance thus will improve estimation accuracy of our P2P model.

2) Distance Factor: Similar to exploring distance matrix d_{ij}, for each data base, we also use the two variations of our P2P model separately derived from setting different distance matrices.

Although CDF of spatial error can describe the estimate errors distribution, it does not illustrate the overall estimate error and is difficult to distinguish a particular distribution among a large number of curves. We thus propose a metric called aggregated error instead of spatial error to have a good view of overall estimate error, which is defined as

$$RelL2_{AG} = \frac{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}(X_{ij}(t) - \hat{X}_{ij}(t))^2}}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}X_{ij}(t)^2}} \quad (31)$$

We vary the value of distance factor s from 0 to 4 with 0.1 as a step, and observe the series of aggregated relative L2 norm errors shown in Fig. 5.

In both cases, aggregated errors of the first variations, where distance matrix elements are all set to be 1, will not be affected by the variation of distance factor s, hence, they are both straight lines in Fig. 1. The aggregated error of pplive-1 reaches the minimum while s is 1.0 and exceeds that of pplive-2 when s is larger than 2.5. It indicates that the distance information has a positive impact on an appropriate distance factor. However, $s = 0.5$ is the best value of distance factor for planet dataset. As is illustrated before, a larger s indicates that the network distance will have more impact on peer selection and data transmission. It is normal and reasonable that different datasets may have distinct preferred values of distance factor s.

C. Performance Comparison

In this subsection, we will compare the estimation performance of P2P model with the other two models, namely Gravity model and Independent connection model.

First we vary the model parameters within the interval $[0, 1]$ and make a comparison among these three models on estimation accuracy in terms of aggregated error. For simplicity, here we only use the pplive dataset and the result is exhibited in Fig. 6. To compare the models fairly, we assume only ingress and egress P2P traffic are available in each model. The x-axis is the free parameter in each model, and the y-axis is the aggregated error.

In Gravity model, there is no free parameter, so the aggregated error is a straight line. In the independent connection model, the free parameter is f, which means the portion of the total bidirectional traffic due to connections that are in the forward direction. f (between 0 and 1) is suggested to set between 0.2 to 0.3 in [8]. In our model, the free parameter is the distance factor of peer selection s.

From Fig. 6, we learn that the performance of Independent connection model largely depends on parameter f. Its aggregated error can be either lower than that of the gravity model or extensively large beyond our consideration. However, it severely depends on parameter. Without any prior distance information, the aggregated error of pplive-1 is only 2% lower than that of the Independent connection model, and 18% lower than that of the Gravity model.

However, after introducing very simple prior distance information, pplive-2 has a great performance gain. The aggregated error of pplive-1 is 32% lower than that of the independent connection model, while 43% lower than that of the Gravity model.

Secondly, we select the best parameter setting of each model and explore the spatial errors of estimation results using these three models. The cumulative distribution of spatial errors using pplive dataset and planet dataset are separately plotted in Fig. 7 (a) and (b).

We can see that in the first case using pplive dataset, roughly 70% of the estimate errors of P2P model are under 0.3. From the trend of CDF curves, P2P model obviously outperforms the other two models.

In the second case using planet dataset, the distribution of

(a) pplive dataset

(b) planet dataset

Fig. 5. Aggregated error with different distance factor s

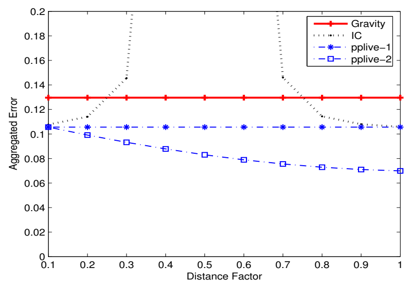


Fig. 6. Aggregated error of three models

spatial errors is similar between the P2P model and Independent connection model, which both outperform the gravity model. However, over 50% of spatial errors using P2P model are under 0.1, whereas the percentage of spatial errors smaller than 0.1 is only 30% using Independent connection model.

Our P2P model thus always exhibits better estimation performance in terms of both spatial error and aggregated error, no matter which dataset to use, than the other two models.

V. Discussion

To the best of our knowledge, this is the first work focusing on P2P traffic matrices estimation problem. Although the performance of our model is well analyzed in Section IV, there remains several issues we should discuss in details.

Q1: Can network operators and ISPs directly benefit from the proposed P2P model?

A1: Yes. Traffic generated by all kinds of P2P applications has occupied a majority of the bandwidth available in Internet. It has long been a desire for network operators and ISPs to improve the efficiency and fairness through traffic engineering approaches.

Based on analysing and modelling characteristics of P2P applications, our P2P model appropriately reflect the distribution of P2P traffic in the form of traffic matrix in the Internet. Owning to the inherent feature of P2P model, network operators and ISPs can have a good picture of P2P traffic distribution with different granularity levels, such as stub-AS level or AS level. As basic input traffic information, a number of measures, such as cache deployment or traffic redirection, could be exerted by network operators and ISPs for traffic management and optimization.

Q2: Is it necessary to generate different P2P models especially for distinct categories of P2P systems and applications?

A2: No. Although our P2P model is derived from the analysis of P2P file share system, it does not mean that our model is only capable of estimating P2P traffic matrices for that exact application. Our model has captured common characteristics among different categories of P2P systems and applications, such as multi-connections and imbalance among peers on uploading volume. Therefore, it is unnecessary to generate a special model for each category of P2P systems and applications.

This is also validated by the evaluation of our model, where both a P2P live streaming dataset and a P2P file sharing dataset are used for performance evaluation. Results have shown that our P2P model is a universal approach to estimate P2P traffic matrices without differentiating application categories.

VI. Related Work

During the past decade many approaches have been proposed to deal with the traffic matrix estimation problem. Several methods, classified as the first generation techniques [4], introduce additional network information and thus turn the problem to be well-constrained. Simple models (e.g. Gaussian, Poisson) have also been applied to reduce the spatial or temporal correlations of OD pairs, which make these methods highly depend on the choice of prior models. The demerit of these methods leads to proposition of the second generation methods.

The gravity model [1], a typical second generation method, utilizes extra information derived from SNMP data. It assumes that the traffic in forward and reverse directions is irrelative. Despite that the gravity model is extensively used for estimating traffic matrices and generating synthetic traffic [9], its assumption on routing schemes between OD pairs has been proved unrealistic in [8]. These second generation approaches provide valuable insight and make a great improvement over the first generation methods, whereas they are incapable of reducing estimate errors.

Recently, a series of methods have been proposed aiming at pushing estimation error rate sufficiently low, such as Principal

(a) pplive dataset

(b) planet dataset

Fig. 7. Relative L2 norm error in space of three models

Components Method [16] and Fanout Method [17]. These methods are classified into the third generation methods in [4], because they commonly rely on traffic data collecting from flow monitors to alleviate the estimate errors. All of these methods assume that when changes in traffic matrix are detected, the flow monitors will turn on for a period of 24 hours. Therefore, the decrease on estimate errors is at the expense of cost increase on traffic measurement. Despite that the third generation methods perform well, as are illustrated in [4], they are not suitable for estimating P2P traffic matrix because of the heavy monitor cost caused by frequent changes of P2P traffic.

The extra information used in the third generation methods is also mainly generated from gravity model. Unlike the gravity model, the independent connection model [8] is a connection oriented model, which assumes the ratio of forward traffic to reverse traffic is constant. This assumption may be unrealistic in the P2P systems. Although P2P applications also exchange data in both directions of the connection, the number of bytes exchanged, however, is partially constrained by link bandwidths. For instance, the ratio of P2P traffic volume from LANs to ADSL is usually larger than that reversely.

VII. Conclusion And Future Work

In this paper, we proposed a novel model to estimate P2P traffic matrices in networks. To the best of our knowledge, this is the first model approach to estimate P2P traffic matrices. Important factors are considered in our model, including the number of peers, the localization ratio of P2P traffic, and the distances among different networks. Here distance can be hop counts or geographic distance accordingly. To validate our model, we have evaluated the performance using both real P2P live steaming traces and file sharing application traces. Evaluation results show that the proposed model outperforms the other two typical models for general traffic matrices estimation, in terms of estimate errors.

Several studies could be carried on as future work, involving considering dynamic features of peers in P2P systems and making more evaluation experiments to further validate our model as well as applying our model to concrete application areas.

References

[1] Zhang Y, Roughan M, Duffield N, Greenberg A. Fast accurate computation of large-scale IP traffic matrices from link loads. *In ACM SIGMETRICS*, 2003.
[2] Medina A, Taft N, Salamatian K, Bhattacharyya S, Diot C. Traffic matrix estimation: existing techniques and new directions. *In ACM SIGCOMM*, 2002.
[3] Gunnar A, Johansson M, Telkamp T. Traffic matrix estimation on a large IP backbone: A comparison on real data. *In IMC*, 2004.
[4] Soule A, Lakhina A, Taft N, Papagiannaki K, Salamatian K, Nucci A, Crovella M, Diot C. Traffic matrices: Balancing measurements, inference and modeling. *In ACM SIGMETRICS*, 2005.
[5] Nucci, A., Sridharan, A., and Taft, N. The problem of synthetically generating IP Traffic Matrices: initial recommendations. *ACM Computer Communication Review*, 35, 3 (2005), 19-32.
[6] Medina A, Salamatian K, Taft N, Matta I, Diot C. A two-step statistical approach for inferring network traffic demands. Technical Report, BUCS-2004-011, Boston: *Computer Science*, Boston University, 2004.
[7] Zhang Y, Roughan M, Lund C, Donoho D. An information-theoretic approach to traffic matrix estimation. *In ACM SIGCOMM*, 2003.
[8] Erramill V, Crovella M, Taft N. An independent-connection model for traffic matrices. *In IMC*, 2006.
[9] Roughan, M. Simplifying the synthesis of Internet Traffic Matrices. *SIGCOMM Computer Communication Review*. 35, 5 (2005), 93-96.
[10] Karagiannis T., Rodriguez P., and Papagiannaki K.. Should internet service providers fear peer-assisted content distribution? *In IMC*, 2005.
[11] Chuan Wu, Baochun Li, Shuqiao Zhao, Characterizing Peer-to-Peer Streaming Flows. *IEEE Journal on Selected Areas in Communications* 25(9): 1612- 1626
[12] PPLive, http://www.pplive.com/.
[13] BitTorrent, http://www.bittorrent.com/.
[14] Saroiu S, Gummadi P, Gribble S. A Measurement Study of Peer-to-Peer File Sharing Systems. *Multimedia Computing and Networking*. 2002.
[15] Xie H, Yang Y R, Krishnamurthy A, et al. P4P: provider portal for applications. *In ACM SIGCOMM*, 2008.
[16] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft. Structural Analysis of Network Traffic Flows. *In ACM SIGMETRICS*, 2004.
[17] K. Papagiannaki, N. Taft, and A. Lakhina. A Distributed Approach to Measure Traffic Matrices. *In IMC*, 2004.
[18] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang. Challenges, Design and Analysis of a Large-scale P2P-VoD System. *In ACM SIGCOMM*, 2008.
[19] Planetlab Testbed, http://www.planet-lab.org/.
[20] GEOIP, http://bloggermap.org/public/geo.ip.php.
[21] Jitendra Padhye, Victor Firoiu, Don Towsley, Jim Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *In ACM SIGCOMM*, 1998.