A Model Approach to the Estimation of Peer-to-Peer Traffic Matrices

Ke Xu, *Senior Member, IEEE*, Meng Shen, Yong Cui, *Member, IEEE*, Mingjiang Ye, and Yifeng Zhong

Abstract—Peer-to-Peer (P2P) applications have witnessed an increasing popularity in recent years, which brings new challenges to network management and traffic engineering (TE). As basic input information, P2P traffic matrices are of significant importance for TE. Because of the excessively high cost of direct measurement, many studies aim to model and estimate general traffic matrices, but few focus on P2P traffic matrices. In this paper, we propose a model to estimate P2P traffic matrices in operational networks. Important factors are considered, including the number of peers, the localization ratio of P2P traffic, and the network distance. Here, the distance can be measured with AS hop counts or geographic distance. To validate our model, we evaluate its performance using traffic traces collected from both the real P2P video-on-demand (VoD) and file-sharing applications. Evaluation results show that the proposed model outperforms the other two typical models for the estimation of the general traffic matrices in several metrics, including spatial and temporal estimation errors, stability in the cases of oscillating and dynamic flows, and estimation bias. 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

1 Introduction

Developing a deep insight into how traffic flows through the network is non-trivial to network operators in network design and management, including traffic engineering, failure recovery, bandwidth provision, etc. The network traffic is usually illustrated by a traffic matrix (TM), which presents traffic volumes between each pair of ingress and egress nodes (e.g., routers) in the network. As basic input information, TM in the context of the Internet is crucial for a wide range of traffic engineering (TE) tasks, such as network planning and load balancing.

Estimation approaches based on partial network information are well accepted to derive traffic matrices because of the excessively high cost of direct online measurement. The estimation problem can be briefly described as follows. Let y be the column vector of measured link loads and x the traffic matrix reorganized as a column vector. The routing matrix is denoted by A, where A_{ij} is 1 if link i serves in the route(s) of node pair $x(j)$, or 0 otherwise. Then the relationship of the three parameters can be expressed as $y = Ax$. We can obtain the link load vector 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 equation above is not straightforward. Since the number of node pairs is much larger than that of links, the matrix A is therefore less than full rank, making the fundamental problem an ill-posed system.

Researchers have proposed a variety of methods and models in recent years to make a more convenient and precise estimation. In [2] both the methods and the models are well summarized. These works mainly focus on the estimation of matrices for *general* traffic regardless of the type of traffic carried over the network.

In the past decade, diverse P2P systems and applications have gained tremendous popularity, which leads to the result that P2P traffic accounts for a major fraction of the Internet traffic [14]. The large volume of P2P traffic significantly increases the load on the Internet, making networks more vulnerable to congestion and failure, and hence brings new challenges to the efficiency and fairness of networks. There has long been a desire for Internet Service Providers (ISPs) to obtain P2P traffic matrices so as to improve overlay routing schemes in a more friendly way for both users and network operators. Existing models designed for general traffic (e.g., the gravity model [3]) fail to capture the features of P2P traffic, leading to undesirable estimation errors for P2P traffic. Therefore, we argue that a model designed especially for estimating P2P traffic is needed and greatly useful.

In this paper, we propose a model to estimate P2P traffic matrices based on a close analysis of the traffic characteristics in P2P systems. To capture the critical properties of the P2P traffic, we take the following physically meaningful factors into consideration. Firstly, the number of peers is considered because, intuitively, networks with more peers might have larger volumes of P2P traffic. Another factor is the traffic localization ratio, which covers the internally exchanged portion of P2P traffic. Last but not least, the distance between different networks is also considered,

- *K. Xu is with the Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science, Tsinghua University, Beijing, China. E-mail: xuke@tsinghua.edu.cn.*
- *M. Shen, Y. Cui, and Y. Zhong are with the Department of Computer Science, Tsinghua University, Beijing, China. E-mail: shenmengnetlab@gmail.com; cuiyong@tsinghua.edu.cn; victorzhyf@gmail.com).*
- *M. Ye is with the Research and Development Center, Yahoo, Beijing, China. E-mail: yemingjiang@gmail.com.*

Manuscript received 9 Apr. 2012; revised 2 July 2013; accepted 9 July 2013. Date of publication 23 July 2013; date of current version 21 Mar. 2014.
Recommended for acceptance by M. Guo.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2013.179

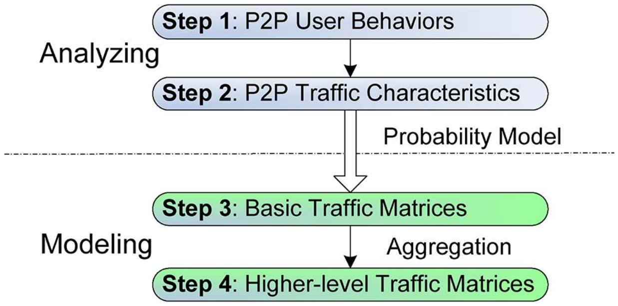


Fig. 1. Methodology of modeling peer-to-peer (P2P) traffic matrices.

which can precisely reflect the peer selection strategy of the concerned system.

Using real P2P traffic datasets derived from a P2P video-on-demand (VoD) system and a P2P file-sharing application, we explore how parameters in the P2P model affect the estimation accuracy. To the best of our knowledge, this is the first work that deals with the estimation of P2P traffic matrices. Therefore, we also evaluate the estimation accuracy of our model through a comparison with two typical models proposed for general traffic matrices, namely the gravity model [3] and the independent-connection (IC) model [6]. Evaluation results show that the newly proposed P2P model outperforms the other two models in several metrics, including spatial and temporal estimation errors, stability in the cases of oscillating and dynamic flows and estimation bias.

The rest of this paper is organized as follows. We illustrate the methodology of modeling P2P traffic matrices in Section 2. In Section 3, the model for the estimation of P2P traffic matrices is proposed, which can be applied to deriving P2P traffic matrices with different aggregation levels. We evaluate the estimation accuracy of our model by using real P2P traffic traces in Section 4. After a brief summary on the related work in Section 5, we conclude this paper in Section 6. Compared with the conference version [1], this paper presents more analytical case studies to verify the generalization of P2P model over diverse P2P applications, and employs more experimental results to evaluate the model performance.

2 Methodology

In this paper, we are dedicated to deriving an accurate model to estimate P2P traffic matrices. Several models have been proposed recently for general traffic estimation [2]. However, existing approaches can not be directly applied to the estimation of P2P traffic matrices, because the high elasticity feature of P2P traffic makes these approaches suffer from either unrealistic assumptions or high measurement costs. Therefore, a precise model should be developed based on a careful study of P2P systems.

In P2P systems, traffic is generated mainly due to the uploading and the downloading process among individual users (also called **peers**). Thus, we can derive an overall picture of traffic interaction by exploring user behaviors, which might be further analyzed to discover some statistical characteristics that are conducive to modeling P2P traffic matrices.

Accordingly, the methodology of modeling P2P traffic matrices could be illustrated by 4 steps grouped into two phases, namely the analyzing and the modeling phase, as shown in Fig. 1. By analyzing user behaviors in P2P systems (Step 1), we can have an insight deep into the characteristics of the traffic exchanged among peers (Step 2). Then, a probability model is applied to getting basic traffic matrices at the level of individual peers (Step 3), based on which we can model higher-level traffic matrices through aggregation (Step 4). We will illustrate the detailed analyzing phase in this section and the modeling phase in next section.

2.1 User Behaviors in P2P Systems

In this subsection, we take BitTorrent [24], a typical and popular P2P file sharing application, as an example to explore user behaviors in P2P systems. In BitTorrent, a large file is divided into smaller data chunks. A peer can simultaneously download multiple chucks from a subclass of its logical neighbors that might be located far from it in terms of geographical distance.

Peer behaviors are quite different in BitTorrent [8]. We can classify peers into three categories according to their contributions to the system:

- **Seeds**: peers that upload a lot of data but never download. In BitTorrent, seeds do not have any bias on choosing which neighbor(s) to upload data to.
- **Free-riders**: peers that download a lot of data but seldom upload. Free-riders are more likely to reject the data requests from other peers.
- **Leechers**: peers that not only download but also upload data. In BitTorrent, leechers prefer uploading peers who have uploaded more data to them before.

2.2 Traffic Characteristics in P2P Systems

The working process of BitTorrent is briefly shown in Fig. 2, which can be roughly divided into three phases: neighbor selection, data request and data transmission.

In the neighbor selection phase, a peer newly in the system registers in a centralized server named tracker (Arrow 1) and retrieves a list of partial peers in the same swarm (Arrow 2), which is a group of peers interested in the same file. In the mainstream implementation of trackers, peers in the list are selected randomly without any bias. But recently, many researchers focus on improving locality in this phase, and prefer to select the neighbors

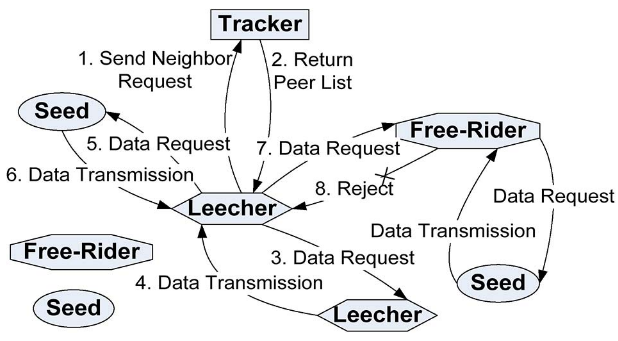


Fig. 2. Working process of BitTorrent.

closer to the requester, such as P4P [20]. The network distance is either measured by peers themselves or provided by ISP-operated services.

In the second phase, the downloading peer will send data requests to its neighbors on the list (Arrow 3, 5, and 7). According to the default setting in BitTorrent, a peer can only concurrently upload data to at most 4 downloading peers, and will reject all received requests when in full uploading service. Leechers will prefer to respond to the data requests from the peers who have uploaded to them before, while free-riders will reject the majority of the received data requests (Arrow 8).

Connections are set up between a host and each of its neighbors who have accepted data requests, and then the data transmission phase begins (Arrow 4 and 6).

Besides the BitTorrent System, similar working processes can also be found in other P2P applications (e.g., PPLive [12]), which are analyzed in Section 1 in the supplementary file which is available in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/179. From the analysis results, several features that are different in P2P systems might affect the traffic volumes among peers, such as the number of concurrent connections and locality-awareness mechanisms [15] in both the neighbor selection and the data request phase. Therefore, we should consider these factors in the modeling phase.

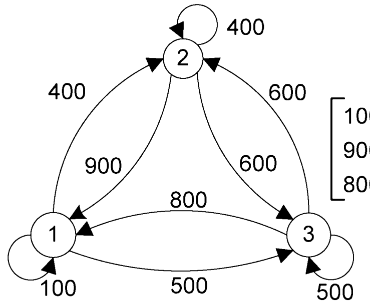
3 Model for P2P Traffic Matrices

In this section, we illustrate the modeling phase to derive a P2P model, based on which an iterative algorithm is designed to estimate P2P traffic matrices.

Generally, an element $X_{ij}(t)$ in traffic matrix $X(t)$ represents the traffic volume from the original node i to the destination node j during a certain time interval t. Here we present a simple topology with three nodes in Fig. 3, where the number beside each arrow denotes the volume of P2P traffic in Mbps flowing in that direction during the time interval t. The P2P traffic matrix is also shown beside the topology. Take Node 1 for example, $X_{11}(t) = 100$, $X_{12}(t) = 400$, and $X_{21}(t) = 900$.

3.1 Modeling Basic P2P Traffic Matrices

We define *basic* P2P traffic matrices as traffic matrices reflecting traffic volumes among individual peers. The basic P2P traffic matrices are difficult to estimate, because individual peers dynamically join and leave the P2P system. To simplify the analysis, we assume that peers remain stable within a certain time interval t, and build up a probability model for basic P2P traffic matrices.



$$\begin{bmatrix} 100 & 400 & 500 \\ 900 & 400 & 600 \\ 800 & 600 & 500 \end{bmatrix}$$

Fig. 3. Simple example of P2P traffic matrix.

Hereafter we use G_s, G_f, and G_l to denote groups of seeds, free-riders and leechers, respectively. Assume that there are altogether N peers in the network denoted by $h_i (i = 1, \ldots, N)$. Considering two individual peers h_i and $h_j (i \neq j)$, the process of h_i sending data to h_j can be divided 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}. Then assume that the data request rate from h_j to h_i is T_{ji}, and the probability of h_i responding to the data request from h_j is P^r_{ij}. Finally, h_i begins transferring data to h_j with the flow throughput B_{ij}. Therefore, the traffic from h_i to h_j is

$$X_{ij} = P^s_{ji} T_{ji} P^r_{ij} B_{ij}. \tag{1}$$

Now we will go through every parameter in equation (1). For a non-seed peer h_j, it retrieves a list containing L_j neighbors from the tracker. The probability of not getting the specific peer h_i in the ψ-th place of the list is $1 - 1/((N-\psi) * (d_{ji})^s)$, where d_{ji} is the network distance between h_j and h_i. Different metrics of the network distance will be evaluated in Section 4.2. The nonnegative s indicates the locality in neighbor selection. If $s > 0$, trackers will take the network distance into consideration with s as its weight; otherwise, all peers will be treated equally. The probability of not getting h_i, denoted by $\overline{P^s_{ji}}$, can be derived through equation (2)

$$\overline{P^s_{ji}} = \prod_{\psi=1}^{L_j} \left(1 - \frac{1}{(N-\psi)(d_{ji})^s}\right). \tag{2}$$

In BitTorrent, there are millions of concurrent users. However, the number of a single peer's neighbors is limited to 100 (e.g., 50). Therefore, it is reasonable to deduce that $N \gg L_j$ always holds in P2P systems. Then the probability of getting h_i is

$$\begin{aligned} P^s_{ji} &= 1 - \overline{P^s_{ji}} \approx 1 - \left(1 - \frac{1}{N(d_{ji})^s}\right)^{L_j} \\ &\approx 1 - \left(1 - \frac{L_j}{N(d_{ji})^s}\right) = \frac{L_j}{N(d_{ji})^s} \end{aligned} \tag{3}$$

where the first approximation holds because $N \gg L_j$, and the second one is derived with the binomial theorem. Seeds will not download data, so they will not retrieve the neighbor list. Therefore, for both seeds and non-seeds, we have

$$P^s_{ji} = \begin{cases} 0, & h_i \in G_s \\ \frac{L_j}{N(d_{ji})^s}, & h_i \notin G_s. \end{cases} \tag{4}$$

Seeds will not request data, so their data request rates are always equal to 0. For a non-seed h_j, it can concurrently send data requests to M_j of the peers on its neighbor list ($M_j \leq L_j$). Assume its total request sending rate is R_j, then the average data request from h_j to h_i is R_j/M_j. Therefore, T_{ji} is expressed as follows:

$$T_{ji} = \begin{cases} 0, & h_j \in G_s \\ \frac{R_j}{M_j}, & h_j \notin G_s. \end{cases} \tag{5}$$

Since free-riders will reject all data requests, P^r_{ij} is 0 when $h_i \in G_f$. If h_i is a seed, when its service capacity S_i (e.g., the maximum number of individual peers it can serve) is not less than the demand D_i, all data requests will be accepted and hence P^r_{ij} is 1; otherwise, h_i could only meet partial demands, and P^r_{ij} is thus S_i/D_i.

As to leechers preferring to upload data to the peers who have uploaded data to them before, P^r_{ij} is also 1, when S_i is not less than demand D_i. But when S_i is insufficient, P^r_{ij} depends on P^r_{ji}. Based on the analysis above, we can get P^r_{ij} as follows:

$$P^r_{ij} = \begin{cases} 0, h_i \in G_f \\ \min\left(1, \frac{S_i}{D_i}\right), h_i \in G_s \\ \left.\begin{matrix} 1, S_i \geq D_i \\ 0, S_i < D_i \end{matrix}\right\}, h_i \in G_l, h_j \in G_f \\ \left.\begin{matrix} 1, & S_i \geq D_i \\ \frac{P^r_{ji}}{\sum_i P^r_{ji}} \frac{S_i}{D_i}, & S_i < D_i \end{matrix}\right\}, h_i \in G_l, h_j \in G_l. \end{cases} \tag{6}$$

When two peers begin transferring data, the data transmission rate is only relevant to the flow throughput. From the classical TCP performance model [21], we can obtain:

$$\mathrm{TCP_{BW}} = \frac{C * \mathrm{MSS}}{\mathrm{RTT}\sqrt{p}} \tag{7}$$

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 all the parameters in equation (7) can be viewed as a measurement of network distance, B_{ij} is in proportion to $1/(d_{ij})^s$. Assume the bottleneck is the uploading capacity rather than the downloading capacity, which is a common assumption in modeling the performance of P2P systems [8]. The uploading capacity of h_i is allocated to different peers according to the weight $1/(d_{ij})^s$.

Therefore, we can deduce the following expression:

$$B_{ij} \propto \frac{1}{(d_{ij})^s} \Rightarrow B_{ij} = \frac{\frac{1}{(d_{ij})^s}}{\sum_{h_j} 1/(d_{ij})^s} U_i \tag{8}$$

where U_i is the total uploading volume of peer h_i. We can decompose equation (1) for different peer types as shown in Table 1, where a seed and a free-rider have no incoming and outgoing traffic, respectively. The rest non-zero elements in Table 1 can be calculated simply by combining (4), (5), (6), and (8)

$$X^{sf}_{ij} = X^{sl}_{ij} = \frac{R_j L_j}{M_j N} \frac{U_i}{(d_{ji})^s} * \min\left(1, \frac{S_i}{D_i}\right) * \frac{\frac{1}{(d_{ij})^s}}{\sum_{h_j} 1/(d_{ij})^s}$$

$$X^{lf}_{ij} = \frac{R_j L_j}{M_j N} \frac{U_i}{(d_{ji})^s} * \left\{\begin{matrix} 1, S_i \geq D_i \\ 0, S_i < D_i \end{matrix}\right\} \frac{\frac{1}{(d_{ij})^s}}{\sum_{h_j} 1/(d_{ij})^s}$$

$$X^{ll}_{ij} = \frac{R_j L_j}{M_j N} \frac{U_i}{(d_{ji})^s} * \left\{\begin{matrix} 1, & S_i \geq D_i \\ \frac{p^r_{ji}}{\sum p^r_{ji}} \frac{S_i}{D_i}, & S_i < D_i \end{matrix}\right\} \frac{\frac{1}{(d_{ij})^s}}{\sum_{h_j} 1/(d_{ij})^s}.$$

TABLE 1
Basic P2P Traffic Matrix

Peer Types	G_s	G_f	G_l
G_s	0	X^{sf}_{ij}	X^{sl}_{ij}
G_f	0	0	0
G_l	0	X^{lf}_{ij}	X^{ll}_{ij}

3.2 The Aggregation of P2P Traffic Matrices

The basic model for each pair of individual peers has been derived in Table 1. Since there are tremendous peers in P2P systems, the basic model is too loose and inadequate to be directly applied for accurate P2P traffic matrices. Therefore, we employ an aggregation process of user groups as well as traffic matrices in Fig. 4, and try to discover statistical patterns after aggregation. For convenience, we predefine three notations as follows,

- **Aggregation level** k, refers to in how many rounds individual peers are aggregated, initialized as 0.
- **Peer cluster** h^k_i, is the i-th set of peers with certain common features (e.g., within the same geographical region or AS) at the aggregation level k.
- **Peer group** H^k, consists of all peer clusters at the aggregation level k, i.e., $H^k = \{h^k_i | i = 1, \ldots, |H^k|\}$.

According to the above definition, every individual peer h_i in a P2P system can also be viewed as a single peer cluster h^0_i, which is an element of peer group H^0, i.e., $H^0 = \{h^0_i | i = 1, \ldots N\}$. Clusters with aggregation level k can be recursively aggregated to derive new clusters at level $k+1$ as shown in the left part of Fig. 4, until satisfy certain termination rules, e.g., a statistical traffic pattern has been observed.

P2P Traffic matrices at different aggregation levels can also be derived when aggregating P2P users. The element X^k_{ij}, presenting P2P traffic volume from h^k_i to h^k_j, can be calculated by equation (9)

$$X^k_{ij} = \sum_{\forall s, h^{k-1}_s \subset h^k_i} \sum_{\forall t, h^{k-1}_t \subset h^k_j} X^{k-1}_{st}, \qquad \forall k > 0. \tag{9}$$

Let I^k_i and E^k_i be total P2P traffic volumes that h^k_i sends to and receives from all other clusters $h^k_j (i \neq j)$, respectively. Then we have

$$I^k_i = \sum_j X^k_{ij}(i \neq j), \quad E^k_i = \sum_j X^k_{ji}(i \neq j). \tag{10}$$

Assume C^k_i is the amount of traffic exchanged within h^k_i. The total uploading and downloading volumes of P2P

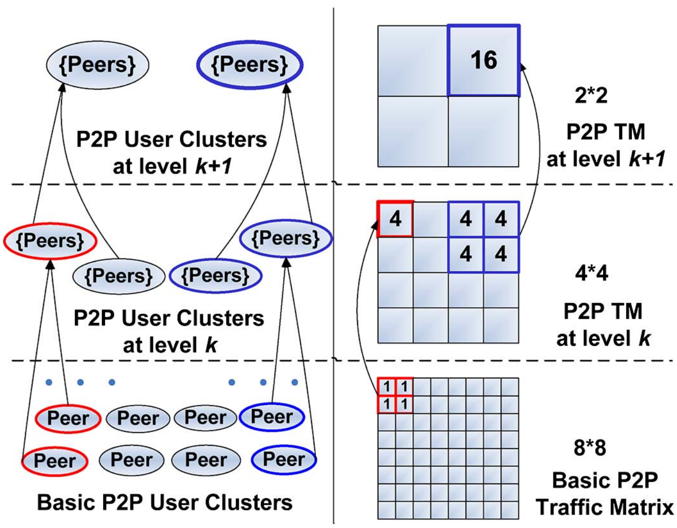


Fig. 4. Aggregation of P2P peers and TMs.

traffic in h_i^k, denoted by U_i^k and D_i^k respectively, can be derived from equation (11)

$$U_i^k = I_i^k + C_i^k, \quad D_i^k = E_i^k + C_i^k. \tag{11}$$

In the example shown in Fig. 3, assume the aggregation level $k = 2$, and the total uploading and downloading volumes of P2P traffic of Node 1 (i.e., h_1^2), which are denoted by U_1^2 and D_1^2 respectively, are 1000 and 1800.

3.3 Modeling High-Level P2P Traffic Matrices

In this subsection, we will show statistical characteristics of P2P traffic among different peer clusters after aggregation, following which the model to estimate P2P traffic matrices will be formally presented.

Through the aggregation process in equation (9), we can derive the P2P traffic matrix at level $k (k > 0)$ from the basic P2P traffic matrix (i.e., with $k = 0$)

$$X_{ij}^k = \sum_{h_m^0 \in h_i^k} \sum_{h_n^0 \in h_j^k} X_{mn}^0 \tag{12}$$

where X_{mn}^0 denotes the basic P2P traffic matrix derived from equation (1). Note that in equation (12), h_m^0 and h_n^0 are individual peers at level 0.

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

Considering the cluster h_i^k with the population of peers $|h_i^k|$, the population ratio of h_i^k over the total number of peers in the system is denoted by μ_i^k (i.e., $\mu_i^k = |h_i^k|/N$). P_{ji}^s is the probability of containing peers belonging to h_i^k on the neighbor lists of peers in cluster h_j^k. Therefore, P_{ji}^s should be proportional to the population ratios of these two clusters μ_i^k and μ_j^k, and inversely proportional to their network distance d_{ij}

$$P_{ji}^s \propto \frac{|h_i^k| \left|h_j^k\right|}{(d_{ij})^s} \propto \frac{\mu_i^k \mu_j^k}{(d_{ij})^s}. \tag{13}$$

For T_{ji}, the request rate of h_j^k is proportional to the total downloading capacity of h_j^k, and the response probability P_{ij}^r is proportional to the uploading capacity of h_i^k, as illustrated in equation (14)

$$T_{ji} \propto D_j^k, \quad P_{ij}^r \propto U_i^k. \tag{14}$$

For h_i^k and h_j^k, the flow throughput between them is inversely proportional to their network distance

$$B_{ij} \propto \frac{1}{(d_{ij})^s}. \tag{15}$$

Then we get the model to estimate P2P traffic matrix at level k shown as below

$$X_{ij}^k = K \frac{\mu_i^k \mu_j^k}{(d_{ij})^s} U_i^k D_j^k \tag{16}$$

where K is a constant to adjust the estimation scale. When considering the time series and ignoring the superscript k, we can finally derive the P2P model shown as follows:

$$X_{ij}(t) = K \frac{\mu_i(t)\mu_j(t)}{(d_{ij})^s} U_i(t) D_j(t). \tag{17}$$

The parameters in the P2P model have their physical meanings. For instance, the distance matrix d_{ij} represents the network distance between each two node (clusters) pairs in terms of AS hop counts or geographic distance. The locality factor s indicates the importance of the network distance on the estimation results.

3.4 Estimating P2P Traffic Matrices

In this subsection, we will show how ISPs can apply the P2P model in equation (17) to the estimation of P2P traffic matrices.

For each time interval t, the total uploading volume $U_i(t)$ and downloading volume $D_i(t)$ of the peer cluster h_i (the superscript aggregation level k is omitted) can be inferred by the total ingress and egress P2P traffic volumes, which can be measured at edge routers. By applying equation (11), we have

$$\begin{aligned} U_i(t) &= \frac{I_i(t)}{\frac{I_i(t)}{U_i(t)}} = \frac{I_i(t)}{1 - C_i(t)/U_i(t)} = \frac{I_i(t)}{1 - \alpha_i(t)} \\ D_i(t) &= \frac{E_i(t)}{\frac{E_i(t)}{D_i(t)}} = \frac{E_i(t)}{1 - C_i(t)/D_i(t)} = \frac{E_i(t)}{1 - \beta_i(t)} \end{aligned} \tag{18}$$

where $\alpha_i(t) = C_i(t)/U_i(t)$ and $\beta_i(t) = C_i(t)/D_i(t)$ are ratios of the local P2P traffic over the total uploading and downloading traffic of cluster h_i, respectively. Then the P2P model in equation (17) can be rewritten as

$$X_{ij}(t) = K \frac{\mu_i(t)\mu_j(t)}{(d_{ij})^s} \frac{I_i(t)}{1 - \alpha_i(t)} \frac{E_j(t)}{1 - \beta_j(t)}. \tag{19}$$

The exact population ratios (i.e., $\mu_i(t)$ and $\mu_j(t)$) are generally difficult to achieve for an ISP. We now explain how the P2P model works without knowing this information. Assume parameters $\alpha_i(t)$ and $\beta_j(t)$ are also unknown in advance, so we treat them together with $\mu_i(t)$ and $\mu_j(t)$. By replacing X_{ij} in equation (10) with the one in equation (19), we can get:

$$\begin{aligned} K \frac{\mu_i(t)}{1 - \alpha_i(t)} \sum_{j (j \neq i)} \frac{\mu_j(t) E_j(t)}{(1 - \beta_j(t))(d_{ij})^s} &= 1, \quad \forall i \\ K \frac{\mu_i(t)}{1 - \beta_i(t)} \sum_{j (j \neq i)} \frac{\mu_j(t) I_j(t)}{(1 - \alpha_j(t))(d_{ij})^s} &= 1, \quad \forall i. \end{aligned} \tag{20}$$

Let $\upsilon_i(t) = \sqrt{K} \frac{\mu_i(t)}{1 - \alpha_i(t)}$ and $\nu_i(t) = \sqrt{K} \frac{\mu_i(t)}{1 - \beta_i(t)}$. Then equation (20) can be rewritten as

$$\upsilon_i(t) \sum_{j (j \neq i)} \frac{\nu_j(t) E_j(t)}{(d_{ij})^s} = 1, \quad \forall i \tag{21a}$$

$$\nu_i(t) \sum_{j (j \neq i)} \frac{\upsilon_j(t) I_j(t)}{(d_{ji})^s} = 1, \quad \forall i. \tag{21b}$$

If we can solve $\upsilon_i(t)$ and $\nu_i(t)$ for all cluster h_i, the desired traffic matrix in equation (19) can be calculated by the following equation:

$$X_{ij}(t) = \frac{\upsilon_i(t)\nu_j(t)I_i(t)E_j(t)}{(d_{ij})^s}. \tag{22}$$

In order to derive $\upsilon_i(t)$ and $\nu_i(t)$, we develop an iterative Algorithm 1 based on equation (21), which requires four input parameters in each fixed time interval t, including the ingress P2P traffic vector $I(t)$, the egress P2P traffic vector $E(t)$, the distance matrix d_{ij} and the locality factor s. The values for the last two parameters are configured either according to physical meanings or through a parameter learning process presented in Section 2.2 in the supplementary file available online.

At the beginning of Algorithm 1, vector $\nu(t)$ is initialized as an iterative seed, and then new values of $\upsilon(t)$ (Line 4-6) and $\nu(t)$ (Line 7-9) are calculated repeatedly in turn according to equation (21), until the gap between ν and its previous value ν' is smaller than a threshold $\theta_{\text{threshold}}$ (Line 10-12). As the P2P traffic volume is usually measured in Gbps, the estimation precision with $\theta_{\text{threshold}} = 10^{-5}$ can reach (by measurement unit) Kbps. The output of Algorithm 1 can be directly used for the estimation of P2P traffic matrix as shown in equation (22).

Algorithm 1: *Iterative Algorithm*

Input: $I(t), E(t), d_{ij}, s$
Output: $\upsilon(t), \nu(t)$

1 Initialize $\nu(t) = [0.1, \ldots, 0.1]$ as an iterative seed, and $\mathcal{N}$ as the number of elements in $I(t)$.
2 **while** *true* **do**
3     $\nu' = \nu$
4   **for** $i = 1$ *to* $\mathcal{N}$ **do**
5         $\upsilon_i = 1/\sum_{j=1}^{\mathcal{N}} \frac{\nu_j E_j}{(d_{ij})^s}(j \neq i)$ in Equation (21a)
6   **end**
7   **for** $i = 1$ *to* $\mathcal{N}$ **do**
8         $\nu_i = 1/\sum_{j=1}^{\mathcal{N}} \frac{\upsilon_j I_j}{(d_{ji})^s}(j \neq i)$ in Equation (21b)
9   **end**
10   **if** $\parallel \nu - \nu' \parallel < \theta_{threshold}$ **then**
11   break;
12   **end**
13 **end**

3.5 Summary

The P2P model we derive here captures the common features of diverse types of P2P applications, and thus is independent of a specific P2P application or system.

The data pieces scheduling policy (e.g., the rarest-first chunk scheduling in BitTorrent), which determines which part(s) of the content should be retrieved first from neighbors, differs in different types of applications and will affect the system-wide throughput. The overall throughput under a concrete scheduling policy can be captured by the uploading and downloading traffic volumes in the P2P model.

Original implementations of P2P systems are locality-agnostic, which causes many inter-ISP traffic and other drawbacks. Therefore, the locality-awareness of P2P systems has attracted much research attention [15], [20]. Due to the diversity of proposed locality-awareness mechanisms, we employ the locality factor s in the P2P model to reflect the overall locality level.

A detailed discussion on these issues is shown in Section 4 of the supplementary file available online.

4 Model Evaluation

In this section, we evaluate the accuracy of our model in estimating Peer-to-Peer traffic matrices. As illustrated above, estimation accuracy partially depends on the choice of parameter values. We thus firstly investigate the influence of parameters on the estimation accuracy of the P2P model in Section 4.2. Since this is the first work focusing on estimating P2P traffic matrices, we can only refer to the existing models proposed for estimating general traffic as benchmarks. Here we select two typical general models, namely the gravity model [3] and the independent-connection (IC) model [6], because the former is widely employed by the research community [22] while the latter makes improvements on the former's assumption. Therefore, in Section 4.3, we analyze estimation results of the P2P model via the comparison with those two models.

4.1 Evaluation Datasets and Metrics

Datasets: Although public traces of general traffic matrices are available online [22], those of P2P traffic are not found yet. Therefore, in our study we collect two P2P traffic datasets to evaluate the performance of our model, which are referred to as `pplive` and `planetlab` dataset, respectively.

The `pplive` dataset is a single ISP-level traffic matrix for PPLive [25], which consists of P2P VoD traffic volumes among six different ISPs in China. The `planetlab` dataset is a series of cluster-level traffic matrices collected by running CTorrent [26] on 289 PlanetLab [27] hosts (i.e., peers) for three consecutive days in Spring, 2011. By analyzing and aggregating traffic traces collected from these hosts, we get a series of $11 * 11$ traffic matrices with setting 1 hour as the time interval. Each of the 11 clusters corresponds to a node in the Abilene backbone. A straightforward aggregation of these hosts is to take all the hosts with the same AS number as an individual cluster. However, the fact that all hosts cover more than 90 ASs makes the derived AS-level traffic matrices too sparse to use. Therefore, we aggregate ASs further based on their geographic locations and finally get 11 clusters. More detailed collection and analysis of these datasets are described in Section 3.1 of the supplementary file available online.

Metrics: For each time interval t, an estimation value will be generated for each origin-destination (OD) flow (i, j), which is denoted by $\hat{X}_{ij}(t)$. To evaluate the estimation accuracy, we leverage the relative L2 norm [2] and define three error metrics by using the $X_{ij}(t)$ and $\hat{X}_{ij}(t)$, including spatial, temporal and aggregated errors.

With the spatial error, we can obtain an error metric for each OD flow (i, j) to summarize estimation errors over its

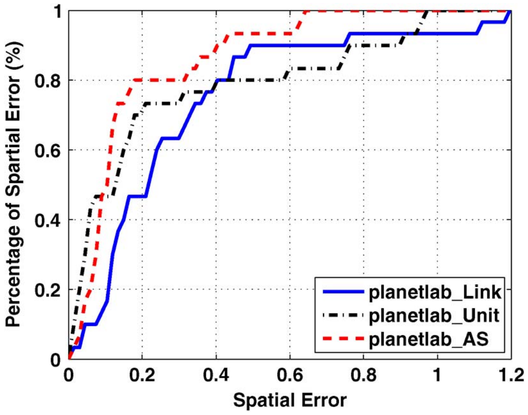


Fig. 5. Cumulative distribution functions (CDFs) of spatial errors vs. the distance matrix d_{ij}.

lifetime (i.e., for all possible t), which is formulated as

$$RelL2_{SP}(i,j) = \frac{\sqrt{\sum_{t=1}^{T}\left(X_{ij}(t) - \hat{X}_{ij}(t)\right)^2}}{\sqrt{\sum_{t=1}^{T} X_{ij}(t)^2}}. \quad (23)$$

The temporal error provides us an error metric that summarizes estimation errors of all the OD flows during a certain time interval t, which is defined as follows:

$$RelL2_{TP}(t) = \frac{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(X_{ij}(t) - \hat{X}_{ij}(t)\right)^2}}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} X_{ij}(t)^2}}. \quad (24)$$

In order to obtain an error metric indicating the overall estimation error of all the OD flows over all time intervals, we combine the spatial and the temporal error with the following equation deriving the aggregated error

$$RelL2_{AG} = \frac{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}(X_{ij}(t) - \hat{X}_{ij}(t))^2}}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T} X_{ij}(t)^2}}. \quad (25)$$

4.2 Evaluation of Parameters in the P2P Model

The estimation accuracy of the P2P model mainly depends on two parameters, namely the distance matrix d_{ij} and the locality factor s. In this subsection, we will use the `planetlab` dataset to investigate the impact of these parameters on the performance of the P2P model. Similar evaluation results are observed for `pplive` dataset. They are described in Section 3.2 of the supplementary file available online.

For ease of the comparison and analysis, we derive three variations of the P2P model by configuring d_{ij} with three different sets of values, which are referred to as `planetlab_Unit`, `planetlab_Link`, and `planetlab_AS`, respectively. Every element of d_{ij} in the first variation is 1; in the second variation, d_{ij} is set to be the shortest link hops of peer cluster pair (i, j) in the Abilene topology, while d_{ij} in the last variation is the average AS-hops of all the pairs of the individual peers belonging to cluster pair (i, j). The AS-hop count of the path from the source peer to the destination peer is derived based on the traceroute information between these two peers [15]. Therefore, both `planetlab_Link` and `planetlab_AS` are determined according to certain physical meanings.

Distance Matrix: In order to evaluate the distance matrix d_{ij}, we fix the locality factor s to 1, and then plot CDFs of spatial errors over all the flows in Fig. 5. We can find that, when $s = 1$, `planetlab_AS` performs slightly better than the other two variations. Therefore, an appropriate setting of the distance matrix d_{ij} will improve the estimation accuracy of the P2P model.

Locality Factor: In order to have a high-level picture on how the locality factor s affects the estimation accuracy, we iteratively change the value of s and observe the corresponding aggregated errors shown in Fig. 6. Considering that $s = 0$ makes $(d_{ij})^s$ always be 1 for any d_{ij}, we therefore vary s from 0.1 to 4 with 0.1 as a step. Aggregated errors of `planetlab_Unit` will not be affected by the variation of s and thus form a straight line. The aggregated errors of `planetlab_Link` and `planetlab_AS` have similar trends, which reach the minimum value of 0.06448 and 0.06433 when $s = 0.4$ and $s = 0.6$, respectively. However, `planetlab_AS` exhibits more gradual increases than `planetlab_Link` as s enlarges.

4.3 Performance Comparison with Existing Models

In this subsection, we will compare the performance of the P2P model with the gravity and the IC model. Since `pplive` is a single traffic matrix within one time interval, we only present comparison results using the `planetlab` dataset.

4.3.1 Preliminary

To make a fair comparison of these three models, we first make sure that every model is configured with appropriate parameters that lead to their respective best performance in terms of estimation accuracy.

In the gravity model, there is no free parameter for adjustment. The adjustable parameter f in the IC model is recommended to be between 0.2 and 0.3 [6], denoting the fraction of the forwarding traffic volume over the total

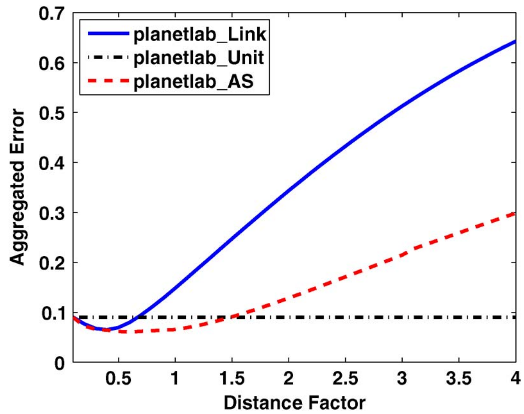


Fig. 6. Aggregated errors vs. the locality factor s.

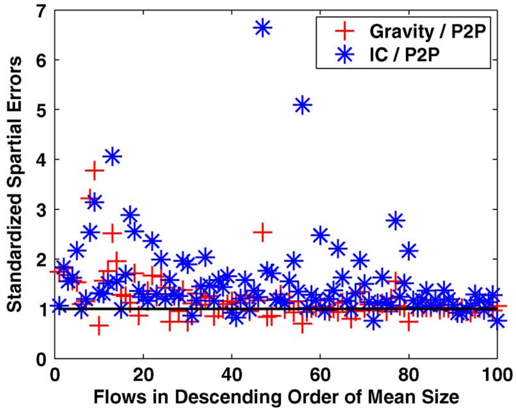


Fig. 7. Spatial errors for three models (x-axis is the flow id).

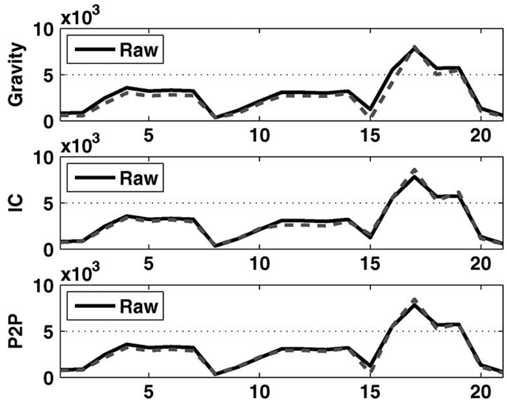


Fig. 9. Dynamics in the flow with the largest mean size.

bidirectional traffic. Here we set $f = 0.2$. For the P2P model, according to Fig. 6, we use the planetlab_AS variation with the locality factor $s = 0.6$.

4.3.2 Spatial Error and Temporal Error

We leverage the spatial and the temporal error to compare the three models in the flow and the time dimension, respectively.

In Fig. 7, we separately plot spatial errors of the gravity and the IC model for each flow divided by errors of the P2P model. The x-axis represents the flow index in the order from the largest to the smallest based on their mean volumes, and the y-axis is the spatial error for each flow standardized by the corresponding value of the P2P model. In this plot, we include all the flows consisting of the top 96 percent of the total load in the traffic matrix. Points above the black base line (i.e., Line $y = 1$) indicate higher spatial errors than those of the P2P model, and vice versa. The errors of the gravity and the IC model are on average 1.21X and 1.54X higher than those of the P2P model, respectively. For flows with a larger size, points are more likely to deviate from the baseline, which implies that the P2P model performs especially better than the other two models for larger flows.

We plot the standardized temporal errors of the gravity and IC models over those of the P2P model in Fig. 8, which totally consists of 21 slots with 1 hour as an interval. The majority of points are above the black base line, and temporal errors of the gravity and IC models are on average 1.33X and 1.83X higher.

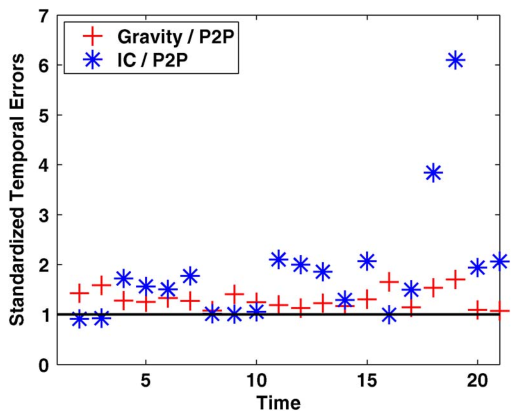


Fig. 8. Temporal errors for three models.

From Figs. 7 and 8, we can find that the P2P model outperforms the other two models in terms of both spatial and temporal errors. In addition, an interesting phenomenon is that when dealing with the P2P traffic, the IC model experiences larger spatial and temporal errors than the gravity model, which differs from the case in estimating general traffic matrices. This also confirms our motivation to propose the P2P model. That is, a model that performs well in estimating general traffic matrices is not necessarily suitable in the context of P2P traffic. An alternative way of viewing spatial and temporal errors, apart from being standardized by the P2P model, is to plot their Cumulative Distribution Functions (CDFs), which are presented in Figs. 4 and 5 in the supplementary file available online.

4.3.3 Handling Dynamic Changes in OD Flows

Due to a variety of reasons, such as network congestions and facility failures, OD flows in networks are fairly dynamic. We also find a sharp oscillation over the entire life time in the planetlab dataset. Therefore, the adaptive ability of a model to deal with dynamic traffic should be highlighted and well evaluated.

We firstly look at the case of an OD flow experiencing weak oscillations. Here we choose the flow with the largest mean size and the oscillations are in the same order of magnitude. The original and the estimated value are plotted in Fig. 9, from which we can see that all of the three models could handle this type of oscillations. However, the P2P model can track the detail extremely well, since over the entire lifetime the gap between the original and the estimated flow in the P2P model is smaller than that of the other two models.

Then we select the OD flow with the largest variance in size to explore the case of a flow experiencing sharp oscillations. The performance of the three models is shown in Fig. 10, where y-axis is over two orders of magnitude. The same with the last case, every model can roughly handle the dynamic changes in the OD flow, whereas the tracking result is different. For the gravity model, the gap between the original and the estimated flow appears in the 1st, 2nd, 3rd, and 13th interval. As to the IC model, the apparent gap

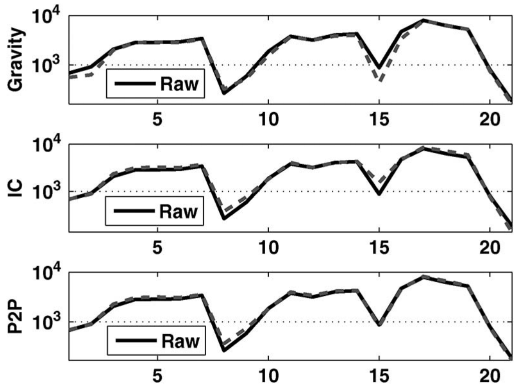


Fig. 10. Dynamics in the flow with the largest variance.

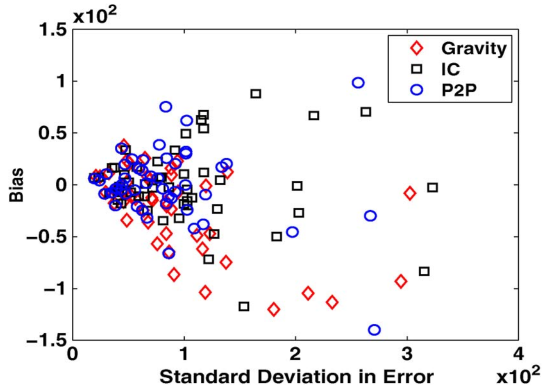


Fig. 12. Estimation bias vs. standard deviation.

exits during the 8th and 13th intervals. In the case of the P2P model, the gap only appears when $t = 8$. In particular, the P2P model handles the sharp oscillation in the 13rd time interval extremely well.

4.3.4 Estimation Bias

Motivated by [2], we finally explore the estimation bias of the three models. Based on the above analysis, we can see that although every model can handle sharp oscillations, they exhibit different estimation results including over-estimation (e.g., the gravity model in the 13th interval in Fig. 10) and under-estimation (e.g., the IC model in the 13th interval in Fig. 10).

The estimation bias of an OD flow (i, j) is a constant value between the real value $X_{ij}(t)$ and the estimation value $\hat{X}_{ij}(t)$ over all time intervals, which can be computed as follows:

$$bias_{ij} = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{X}_{ij}(t) - X_{ij}(t)\right). \quad (26)$$

The bias is shown in Fig. 11 for each model, where the x-axis represents the OD flow in the descending order according to their mean sizes, and the y-axis is the computed bias with the unit of 10^2. In general, three models are biased, and the amount of bias increases with the size of an OD flow. Although both positive and negative biases are shown for three models, the majority of the biases for the P2P model is within the interval $[-1, 1]$, comparing with a wider interval $[-3, 1]$ for the gravity model.

Since the accuracy of an estimator is influenced by both sample bias and variance, we explore the relationship between these two factors for the three models. Here we leverage the definition of sample standard derivation of a model for an OD flow (i, j), as given below

$$ErrStd_{ij} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(err_{ij}(t) - bias_{ij}\right)^2} \quad (27)$$

where $err_{ij}(t) = \hat{X}_{ij}(t) - X_{ij}(t)$.

The sample bias of each flow against its standard derivation is plotted in Fig. 12, for the three models. We can find that the bias and the variance of the gravity model are correlative. That is, the value of standard deviation increases as the bias enlarges. For example, the deviation is roughly lower than 1.5 when the bias varies within $[-0.5, 0.5]$ (both units are 10^2). As to the IC model, the plotted points are too dispersedly distributed to observe the regular relationship between those two factors. Among these three models, the P2P model maintains both relatively low bias and low variance, which tends to both estimate the flow mean well over long time slots and track changes in the flow with frequent variations. More evaluation results of the P2P model are presented in Section 3 of the supplementary file available online.

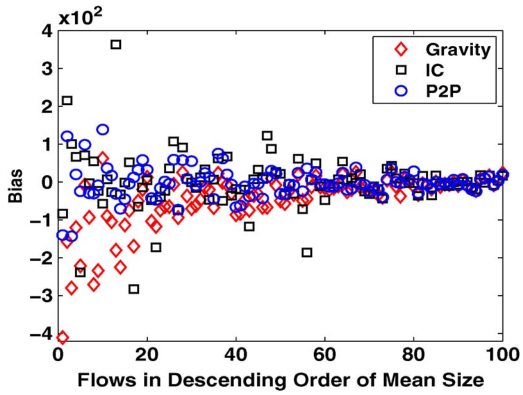


Fig. 11. Bias vs. mean flow size.

5 RELATED WORK

Traffic Matrices: Many estimation methods has been developed to estimate traffic matrices based on available but inadequate information (e.g., link traffic measurements) [3], [4], [5], [6]. The extensively used gravity model [3] assumes that the traffic in the forward and the backward direction is irrelative. The amount of traffic from node i to node j is proportional to the amount of traffic departing the network at j divided by the total amount of traffic departing the entire network. However, this assumption has been proved unrealistic in [6]. The IC model [6] is a connection-oriented model, which assumes a constant ratio of the forwarding traffic over the total bidirectional traffic of a connection.

These models are proposed for general traffic, whose assumptions are invalid for P2P applications. For instance, the assumption in the IC model is unrealistic because the ratio of P2P traffic in the forwarding direction over the total traffic varies a lot as a result of diverse peer behaviors. In this paper, we focus on designing a model to estimate P2P traffic matrices with improved accuracy. To this end, several common characteristics in different P2P applications are incorporated into the P2P model, such as the network distance between each two pairs of peer clusters, and the locality factor indicating the system-wide traffic locality ratio.

P2P Systems: Recent research on P2P systems can be roughly classified into two categories: measurement and improvement. The measurement studies in the first category try to understand how various P2P applications perform, including file-sharing systems [7], live-streaming systems [9], [10], video-on-demand (VOD) systems [11], [12], and so on.

Extending the measurement studies, research in second category is dedicated to improving the effectiveness and ISP-friendliness of P2P systems. Qiu and Srikant [13] develop a built-in incentive mechanism based on the game theory for BitTorrent. The locality-agnostic feature in the overlay network of P2P systems leads to many unnecessary inter-ISP traffic [14] and longer data transmission delays; several solutions [15], [16], [17], [19], [20] have been put forward and evaluated based on either simulation or real-world Internet topologies. Liu *et al.* [15] show that locality-awareness can help the existing P2P solutions significantly reduce the load on the Internet, and achieve shorter downloading time.

With the proposed P2P model, we reveal common traffic patterns that are independent of specific P2P systems, from which both P2P system developers and network operators can benefit. For example, the P2P model can be applied to the deployment of the P2P caches [16].

6 Conclusion and Future Work

In this paper, we first develop a deep insight into user behaviors and traffic characteristics of P2P systems, and then propose a novel model to estimate P2P traffic matrices. In order to better reflect the features of P2P traffic, we consider several important factors, including the localization ratio of P2P traffic and the network distance. To the best of our knowledge, this is the first model approach to the estimation of P2P traffic matrices. Evaluation results based on real traffic datasets show that the proposed model outperforms the other two typical models for general traffic matrices estimation, in terms of estimation errors, stability in the cases of oscillating and dynamic flows and estimation bias.

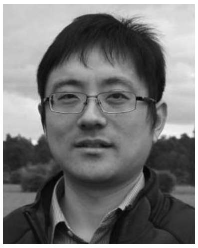
Several studies could be carried on as future work, including carrying on more evaluation experiments to further validate our model and applying the P2P model to concrete application areas, such as the cache deployment for P2P traffic.

Acknowledgment

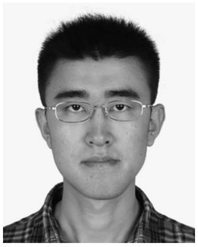
NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] K. Xu, M. Shen, and M. Ye, ''A Model Approach to Estimate Peer-to-Peer Traffic Matrices,'' in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 676-684.
[2] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, ''Traffic Matrices: Balancing Measurements, Inference and Modeling,'' in *Proc. ACM SIGMETRICS*, 2005, pp. 362-373.
[3] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, ''Fast Accurate Computation of Large-Scale IP Traffic Matrices From Link Loads,'' in *Proc. ACM SIGMETRICS*, 2003, pp. 206-217.
[4] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, ''Traffic Matrix Estimation: Existing Techniques and New Directions,'' in *Proc. ACM SIGCOMM*, Aug. 2002, pp. 161-174.
[5] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, ''An Information-Theoretic Approach to Traffic Matrix Estimation,'' in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 301-312.
[6] V. Erramill, M. Crovella, and N. Taft, ''An Independent-Connection Model for Traffic Matrices,'' in *Proc. IMC*, 2006, pp. 251-256.
[7] A.R. Bharambe, C. Herley, and V.N. Padmanabhan, ''Analyzing and Improving a BitTorrent Network's Performance Mechanisms,'' in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1-12.
[8] S. Saroiu, P. Gummadi, and S. Gribble, ''A Measurement Study of Peer-to-Peer File Sharing Systems *Proc. Multimedia Comput. Netw.*, Jan. 2002, pp. 156-170.
[9] C. Wu, B. Li, and S. Zhao, ''Characterizing Peer-to-Peer Streaming Flows,'' *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1612-1626, Dec. 2007.
[10] X. Zhang, J. Liu, B. Li, and T.P. Yum, ''CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming,'' in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2102-2111.
[11] Z. Ma, K. Xu, J. Liu, and H. Wang, ''Measurement, Modeling and Enhancement of BitTorrent-Based VoD System,'' *Comput. Netw.*, vol. 56, no. 3, pp. 1103-1117, Feb. 2012.
[12] Y. Huang, T. Fu, D.M. Chiu, J.C.S. Lui, and C. Huang, ''Challenges, Design and Analysis of a Large-Scale P2P-VoD System,'' in *Proc. ACM SIGCOMM*, Aug. 2008, pp. 375-388.
[13] D. Qiu and R. Srikant, ''Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks,'' in *Proc. ACM SIGCOMM*, Aug. 2004, pp. 367-377.
[14] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, ''Should Internet Service Providers Fear Peer-Assisted Content Distribution?'' in *Proc. IMC*, 2005, p. 6.
[15] B. Liu, Y. Cui, Y. Lu, and Y. Xue, ''Locality-Awareness in BitTorrent-Like P2P Applications,'' *IEEE Trans. Multimedia*, vol. 11, no. 3, pp. 361-371, Apr. 2009.
[16] M. Ye, J. Wu, and K. Xu, ''Caching the P2P Traffic in ISP Network,'' in *Proc. IEEE ICC*, May 2008, pp. 5876-5880.
[17] K. Xu, M. Zhang, J. Liu, Z. Qin, and M. Ye, ''Proxy Caching for Peer-to-Peer Live Streaming,'' *Comput. Netw.*, vol. 54, no. 7, pp. 1229-1241, May 2010.
[18] K. Xu, H. Li, J. Liu, W. Zhu, and W. Wang, ''PPVA: A Universal and Transparent Peer-to-Peer Accelerator for Interactive Online Video Sharing,'' in *Proc. IEEE IWQoS*, 2010, pp. 1-9.
[19] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, ''Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences with LiveSky,'' in *Proc. MM*, Oct. 2009, pp. 25-34.
[20] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, ''P4P: Provider Portal for Applications,'' in *Proc. ACM SIGCOMM*, Aug. 2008, pp. 351-362.
[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, ''Modeling TCP Throughput: A Simple Model and Its Empirical Validation,'' in *Proc. ACM SIGCOMM*, Sept. 1998, pp. 303-314.
[22] Yin Zhang's Abilene TM. [Online]. Available: http://www.cs.utexas.edu/~yzhang/\\research/AbileneTM/
[23] Internet Studies. [Online]. Available: http://www.ipoque.com/sites/default/files/\\mediafiles/documents/internet-study-2008-2009.pdf
[24] BitTorrent. [Online]. Available: http://www.bittorrent.com/
[25] PPLive. [Online]. Available: http://www.pplive.com/
[26] CTorrent. [Online]. Available: http://ctorrent.sourceforge.net/
[27] Planetlab Testbed. [Online]. Available: http://www.planet-lab.org/



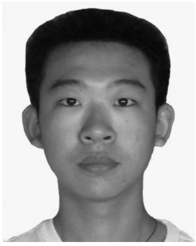
Ke Xu received the PhD degree from the Department of Computer Science of Tsinghua University, Beijing, China, in 2001. Currently, he is a Full Professor in the Department of Computer Science of Tsinghua University. He has published more than 100 technical papers and holds 20 patents in the research areas of next generation Internet, P2P systems, Internet of Things (IoT), network virtualization, and optimization. Dr. Xu is a Member of ACM. He has guest edited several special issues in IEEE and Springer Journals. He is serving as an Associate Editor of Springer journal Networking Science. He is a Senior Member of the IEEE.



Meng Shen received the BS degree in computer science from Shandong University, Shandong, China, in 2009. Currently, he is a PhD candidate in the Department of Computer Science of Tsinghua University, Beijing, China. His research interests include network congestion control, P2P traffic management, and network virtualization.

Yong Cui received the BS and PhD degrees in computer science from Tsinghua University, Beijing, China in 1999 and 2004, respectively. Currently, he is a Full Professor in Tsinghua University and a Co-chair of IETF IPv6 Transition WG Softwire. Having published more than 100 papers in refereed journals and conferences, he also received the Best Paper Awards from ACM ICUIMC 2011 and WASA 2010. Holding more than 40 patents, he is one of the authors in IETF RFC 5747 and RFC 5565 for his proposal on IPv6 transition technologies. His major research interests include mobile wireless Internet and computer network architecture. He is a Member of the IEEE.

Mingjiang Ye received the BS and PhD degrees in computer science from Tsinghua University, Beijing, China, in 2004 and 2009, respectively. Currently, he is a Senior Engineer Manager in Yahoo! His major research interests include P2P network, Internet traffic management, and computational advertising.



Yifeng Zhong received the BS degree in software engineering from Tianjin University, Tianjin, China, in 2011. Currently, he is a PhD student in the Department of Computer Science of Tsinghua University, Beijing, China. His research interests include P2P network and network economics.