Towards an ISP-Compliant, Peer-Friendly Design for Peer-to-Peer Networks*

Haiyong Xie, Yang Richard Yang, and Avi Silberschatz

Computer Science Department, Yale University, New Haven CT 06511, USA
{haiyong.xie,yang.r.yang,avi.silberschatz}@yale.edu

Abstract. Peer-to-peer (P2P) applications are consuming a significant fraction of the total bandwidth of Internet service providers (ISPs). This has become a financial burden to ISPs and if not well addressed may lead ISPs to block or put strict rate limits on P2P traffic. In this paper, we propose a new framework, *PCP*, for designing P2P applications to smoothly fit into the global Internet. In our framework, an ISP decides on how much of its bandwidth is to be allocated to P2P clients, and P2P clients inside the network adopt a peer-friendly algorithm to fairly share the bandwidth. Using the widely-used percentile-based charging model and real traffic traces, we show that an ISP can allocate a large amount of bandwidth dedicated to P2P, without increasing its financial cost. We also show that P2P clients can use the algorithm to fairly share the allocated bandwidth.

1 Introduction

There are increasing concerns on Peer-to-Peer (P2P) applications by Internet Service Providers (ISPs). P2P traffic consumes an increasingly significant fraction of network bandwidth of ISPs. A recent study [7] estimates that the aggregated traffic of all P2P applications contributes to about 60-70% of the traffic in Internet and about 80% of the traffic in the last-mile providers' networks. Thus P2P traffic increases network utilization, and can result in performance degradation to other applications. More importantly, P2P traffic is becoming a financial burden to the ISPs. Most ISPs connect to the Internet through their providers, and pay for the transit services. However, it may well happen that P2P clients in an ISP's network upload a substantial amount of data to their peers in the ISP's provider networks; hence, the ISP may become a content provider and be charged by its network provider unnecessarily. In a recent study [12] on Skype, the authors find that many universities are hosting a large number of Skype super nodes, and becoming potential providers to the rest of the Internet. In such networks, the outbound P2P traffic can be 5 times more than normal inbound traffic [10]. This can incur substantially higher operational cost to the university network. The world-wide extra costs due to P2P traffic are estimated to be in excess of € 500M per annum [7].

* Haiyong Xie is supported by NSF grants ANI-0238038 and CNS-0435201. Yang Richard Yang is supported in part by NSF grants ANI-0238038 and CNS-0435201.

A. Das et al. (Eds.): NETWORKING 2008, LNCS 4982, pp. 375–384, 2008.

However, ISPs do not have effective methods to handle the increasing P2P traffic. First, an ISP can upgrade its infrastructure. However, this may increase its over-provisioning cost substantially; further, the upgraded bandwidth might quickly be consumed by the constantly increasing P2P traffic. Second, an ISP can use P2P caching devices to reduce P2P bandwidth consumption. However, many ISPs are reluctant to get involved in content distribution by some P2P applications due to legal concerns. Third, an ISP may choose to block P2P traffic. However, this may involve complicated legal and publicity issues. Last, an ISP can use traffic shaping devices (*e.g.*, [6,9]) to enforce a hard limit on P2P bandwidth consumption. However, P2P applications have taken countermeasures, for instance, hide themselves by using existing application-layer protocols and using random port numbers and message stream encryption [1]. Such evading techniques not only substantially increase the complexity but also reduce the interoperability and impede the development of P2P applications.

In this paper, we propose a new framework referred to as *Peer Coordination Protocol*[1] (or *PCP* for short). The framework allows ISPs to decides its bandwidth allocated to P2P clients according to its policy constraints, for example, an ISP may want to bound the total extra bandwidth and financial costs incurred by P2P applications. A clear advantage of the framework is that it allow an ISP to explicitly express its policies and constraints on P2P bandwidth usage.

We also design algorithms for an ISP to determine P2P bandwidth allocation without increasing its cost in the widely-used percentile-based charging model. We show that our dynamic approach can allocate a large amount of bandwidth (40-60% of total bandwidth) to P2P applications without increasing its financial cost. Thus, our framework not only gives ISPs explicit control over bandwidth allocation, but also gives P2P clients incentive to adopt the approach since they can obtain a large amount of bandwidth which may not be feasible otherwise. In addition, we design a distributed algorithm for P2P clients inside an ISP network to be peer-friendly and share the allocated P2P bandwidth. Using real Internet traces, we demonstrate the effectiveness of our algorithm.

2 A Framework for ISP-Compliant, Peer-Friendly Design

Consider the scenario where an ISP network is connected to its provider through a peering link. Our focus is how the ISP should allocate the bandwidth of this link, and how P2P clients in the ISP should share the allocated bandwidth efficiently. Although we only consider a single peering link in the setup, our framework can be easily generalized to deal with multiple peering links.

The first component is the algorithm to compute the total bandwidth allocatable to all P2P clients in the ISP network (referred to as *total P2P capacity*). The ISP should communicate the total P2P capacity to P2P clients in its network. One implementation strategy is to use one or multiple capacity allocation servers for this purpose. The ISP's DNS server maintains a unique DNS resource

[1] It is worth of noting that besides P2P, our framework is applicable to other applications as well.

record (*e.g.*, PCP) for these servers. P2P clients should query them through DNS lookups to obtain the total P2P capacity. Common practices such as DNS-based round robin and IP-based query filtering can be used to balance the query load and preserve privacy.

The second component is the peer-friendly algorithm for P2P clients to fairly share the total P2P capacity. This component is necessary since the ISP may not want to be involved in determining how P2P clients should share the allocated capacity. Further, there may exist multiple P2P applications and/or clients in the ISP network, and they may not be aware of each other. Therefore, it is desirable that the capacity sharing algorithm be distributed.

3 Percentile Networks

We now study an ISP-compliant, peer-friendly P2P design for percentile networks where an ISP's transit costs are determined by the widely-used percentile-based charging model. We study two common policies where the ISP allocates P2P capacity so that (1) the additional P2P traffic does not increase the ISP's transit cost; and (2) the total traffic on the transit link does not exceed link capacity. Note that the ISP may impose additional constraints, for example, a threshold on network utilization. Our framework can be extended to handle such constraints.

The percentile-based charging model is a typical usage-based charging scheme currently in use by many ISPs (*e.g.*, [5]). Consider an ISP obtains transit services through a transit link with capacity C to its provider. The ISP's provider records the traffic volumes on the transit link in 5-minute intervals, denoted by v_i for the i-th interval. Let V be the vector consisting of the volumes in a charging period (typically one month) of I intervals. The provider charges the ISP using the q-th percentile of the sorted V (referred to as charging volume).

We denote by x_i and y_i the non-P2P and P2P traffic volume on the transit link in the i-th interval, respectively. Let X and Y be the vectors of non-P2P and P2P traffic volumes. We also denote by K the number of P2P clients in the given ISP's network, and y_i^k the capacity P2P client k uses on the transit link in the i-th interval. Thus $Y^k = \{y_i^k | 1 \leq i \leq I\}$, and $Y = \sum Y^k$. Let $p = \mathtt{qt}(V, q)$ be the charging volume, *i.e.*, the q-th percentile of the traffic volume vector V.

We formulate the aforementioned two ISP policies as follows: (1) for the cost constraint, and (2) for the link capacity constraint.

$$\mathtt{qt}(X + Y, q) = \mathtt{qt}(X, q); \qquad (1)$$

$$\forall i : x_i + \sum_{k=1}^{K} y_i^k \leq C. \qquad (2)$$

3.1 Algorithm to Determine Total Allocated P2P Bandwidth

We next present an algorithm for the ISP to determine the total P2P capacity for each interval (*i.e.*, $p - v_i$). However, p is unknown until the end of a charging

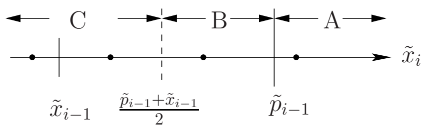

Fig. 1. Network status

for each time interval i in a charging period
    obtain $\tilde{c}_i$ for the current interval
    obtain last interval's network state
    // update division factor
    if case A
        $\tilde{K}_i = \tilde{K}_{i-1} * 2$
    else if case B
        $\tilde{K}_i = \max\{1, \tilde{K}_{i-1} - 1\}$
    else if case C
        $\tilde{K}_i = \frac{\tilde{c}_i}{\tilde{c}_{i-1}} \times \tilde{K}_{i-1}$
    // compute its share of total P2P capacity
    $y_i = \tilde{c}_i / \tilde{K}_i$

Fig. 2. A capacity sharing algorithm

period, and v_i is unknown until the end of an interval i. We can predict p and v_i at the beginning of each interval. To facilitate the prediction, we collect traffic volumes in previous intervals by repeatedly querying the border router through SNMP[2].

We take a *hybrid sliding window* approach to predicting p. Specifically, for the first M intervals in a charging period, the traffic volumes of the last charging period (up to and including the last interval) are used for predicting the charging volume; while for the remaining $I - M$ intervals, the traffic volumes from the first interval to the latest interval in the current charging period are used for prediction. We denote by $\tilde{p}_i$ the predicted charging volume:

$$\tilde{p}_i = \begin{cases} \mathtt{qt}(V[i-I, i-1], q) & \text{for } s \leq i \leq M, \\ \mathtt{qt}(V[s, i-1], q) & \text{for } M < i < s + I, \end{cases} \tag{3}$$

where $s = \lfloor \frac{i}{I} \rfloor * I + 1$ is the first interval in the current charging period.

We take a similar approach to predicting v_i for each interval i. Specifically, a sliding window of the recent N intervals is used for the prediction. We denote by $\tilde{v}_i$ the predicted traffic volume: $\tilde{v}_i = \frac{1}{N} \sum_{j=i-N}^{i-1} v_j$. Similarly we can predict $\tilde{x}_i$. We then estimate the predicted P2P capacity as $\tilde{c}_i = \tilde{p}_i - \tilde{x}_i$ for each interval i. Note that the size of the sliding window is a parameter of the prediction algorithm. However, it cannot be too large; otherwise, the diurnal traffic patterns will be lost in the prediction.

3.2 A Distributed, Peer-Friendly Sharing Algorithm

We next present an algorithm for P2P clients to share the allocated P2P capacity. At the beginning of an interval, each P2P client first queries the ISP to obtain the current P2P capacity c_i and then estimate its fair share. The estimation relies on an aggregated network status and an internal division factor. The latter is an estimation of the total number of P2P clients in the network, denoted by $\tilde{K}_i$.

[2] The router's MIB keeps a counter for the traffic volume v_i in each 5-minute interval i, and this is widely implemented in most of the off-the-shelf routers.

Table 1. Traffic traces and rates (Mbps)

AS	Organization	Traffic Rate	Charging Volume	Total P2P Capacity	%
52	UCLA	47.550	95.614	48.063	50%
55	University of Pennsylvania	39.793	67.227	27.433	41%
111	Boston University	43.314	85.674	42.359	49%
237	Merit Network Inc.	103.772	185.027	81.254	44%
1249	Five Colleges Network	31.189	63.073	31.887	51%
22753	Red Hat Inc.	28.079	64.611	36.531	57%

Network status: We divide the network status into three cases, as shown in Figure 1. We refer to a network in interval $i-1$ as *over-utilized* if $\tilde{p}_{i-1} < v_{i-1}$, and *under-utilized* otherwise. Therefore, case A implies that the network is likely to be over-utilized, while case B and C correspond to a potentially low and high under-utilization, respectively. Note that over-utilization may result in violation of ISP-compliant constraints [3]; therefore, P2P clients should reduce their traffic rates more aggressively in over-utilization states.

Division factor: Each P2P client adjusts its local division factor using the algorithm in Figure 2. Specifically, the algorithm aggressively doubles its division factor when the network is over-utilized, resulting in reducing its share of the total P2P capacity by half. When in case B, the division factor is reduced gradually by one until it reaches the lowest value 1; thus, the P2P node can gradually increase its share of the the total capacity. When in case C, the division factor is updated proportional to the ratio of the P2P capacities in the current and latest interval.

It is particularly desirable that the network is over-utilized as least often as possible, since this reduces the chances of violating ISP-compliant constraints. We can show by Proposition 1 (we omit its proof due to space limit) that the algorithm guarantees that the network is not over-utilized if it is under-utilized in the preceding interval.

Proposition 1. *Assume that the network is under-utilized in interval* $i-1$ *and* $x_i \leq x_{i-1}$. *Assume also that the prediction for the charging volumes is accurate and that the network has a fixed number of P2P nodes. The peer-friendly sharing algorithm guarantees that the network is not over-utilized in the next interval.*

4 Evaluations

We use real Internet traffic traces in our evaluations. The traces were collected from Abilene, containing 5-minute traffic volumes from Nov. 1, 2003 to Dec. 31, 2003. We take the traffic volumes in these traces as non-P2P traffic. We also use the 95th-percentile charging model and the charging period is one month. For brevity, we present the results obtained using the traces of AS 111 only, since others have similar results.

[3] In a q-percentile charging model, the charging volume will increased if the total P2P capacity is over-utilized in more than q percent of intervals.

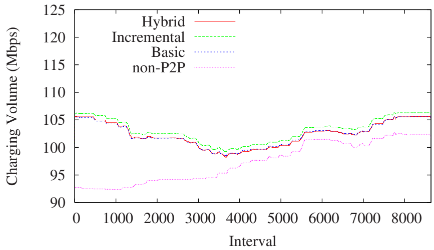

Fig. 3. Charging volume prediction

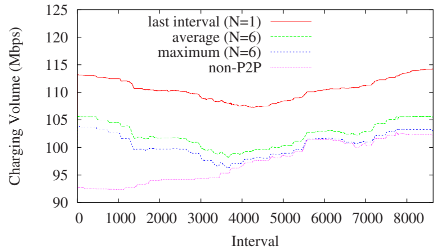

Fig. 4. Traffic volume prediction

We consider three performance metrics. The first is charging volume, from which we can derive the total costs. The second is robustness of the capacity prediction algorithm to traffic pattern changes. The third is efficiency of the capacity sharing algorithm. A desirable algorithm should enable multiple P2P clients to efficiently and fairly share the allocated P2P capacity.

We evaluate the following algorithms of charging volume prediction: (1) the algorithm based on Equation (3), (2) a basic sliding window algorithm that uses the recent I intervals to predict the charging volume, and (3) an incremental sliding window algorithm that uses known volumes in the current charging period only. We also evaluate the algorithms predicting traffic volume as the average and maximum volume in a varying number N recent intervals, respectively.

4.1 Total Available P2P Capacity in Percentile Networks

Table 1 summarizes the results of the available P2P capacity in percentile networks. We observe that all six ASes can allocate 40% – 60% of their bandwidth usage to P2P applications without increasing their bandwidth costs. In other words, these networks can accommodate about the same amount of additional P2P traffic as that of non-P2P traffic at no extra financial costs. This is clearly an evidence that a dynamic policy may be more desirable since it is unlikely that an ISP will *statically* allocate 40% to 60% of their bandwidth to P2P applications.

4.2 Algorithm to Determine the Total P2P Capacity

Figure 3 plots the results of evaluating charging volume prediction algorithms, assuming that traffic volumes are known *a priori*. In order to show the dynamics of the algorithms, we plot the charging volumes computed in all intervals, and only the last point on each curve represents the real charging volume for the complete charging period. We observe that all three algorithms result in higher charging volumes due to prediction errors. In particular, the hybrid/basic and incremental algorithms increase the charging volume by 3% and 6%, respectively. An ISP pays only a small penalty by using these algorithms.

Fig 4 plots the results of evaluating traffic volume prediction algorithms, where we use the hybrid sliding window algorithm to predict the charging volume.

We observe that the last-interval prediction algorithm ($N = 1$) increases the charging volume by 12%, while the maximum and average algorithms ($N = 6$) increase the charging volume by only 1% and 3%, respectively. The results are consistent when N is larger. The last-interval prediction relies too much on the most recent history of traffic rates, thus is likely to have higher prediction errors when the traffic rates fluctuate. The maximum algorithm slightly outperforms the average algorithm. However, we find that it allocates 26.167 Mbps, while the average algorithm allocates 33.210 Mbps (or 27% more), as the P2P capacity. This suggests that the average algorithm is more desirable as a traffic volume prediction algorithm in percentile networks.

4.3 Distributed, Peer-Friendly Sharing Algorithm

Figure 5 plots the results of evaluating the capacity sharing algorithm. We observe that the charging volume increases by 6%. This penalty is due to the prediction error in P2P capacity prediction (as shown in Section 4.2) and underestimation of of the division factor.

Figure 6 illustrates how efficiently the P2P capacity is shared by P2P clients. We observe that P2P clients coordinate themselves to share the allocated P2P capacity. We define the P2P capacity utilization ratio as the ratio between the aggregated P2P traffic volumes and the aggregated P2P capacity (sum of P2P capacity allocated in all intervals). We find that the P2P capacity utilization ratio is 71%; in other words, the two P2P nodes utilize 71% of all available P2P capacity using the capacity sharing algorithm. We also find that the clients generate approximately equal amount of traffic. Although the evaluation is very preliminary, the result suggests that the distributed, peer-friendly sharing algorithm is efficient for P2P clients to fairly share the allocated P2P capacity.

4.4 Robustness to Changing Traffic Patterns

To evaluate the robustness of the proposed algorithms to the Internet traffic dynamics, We repeat the experiments of Section 4.2 using the traces of Dec. 2003, as they have changing traffic patterns.

Figure 7 plots the results of charging volume prediction algorithms. We observe that all three schemes are robust to the traffic dynamics, with only about 9% increase in the final charging volume. We also observe that the hybrid algorithm slightly outperforms the incremental algorithm, and that both of them outperforms the basic sliding window algorithm.

Figure 8 plots the results of the algorithm to determine the total P2P capacity. We observe that the last-interval prediction is *not* robust due to the same reason described in the preceding subsection. while the average and maximum algorithm increase the charging volume by 14% and 7%, respectively. This suggests that the latter two algorithms are more desirable for ISPs having non-regular diurnal traffic patterns to predict the total P2P capacity.

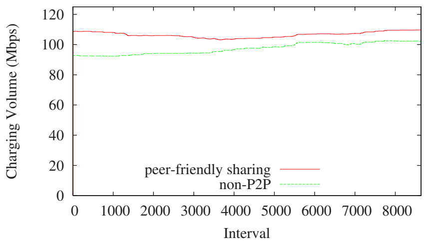

Fig. 5. Charging volume when using capacity sharing algorithm

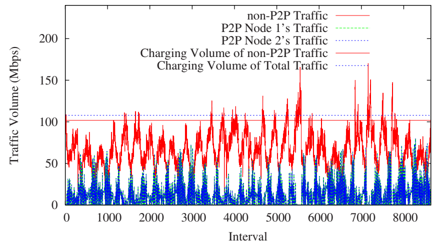

Fig. 6. P2P capacity utilization when using capacity sharing algorithm

5 Discussions

We first consider the problem that a P2P client has to make an initial estimation about the division factor when it first becomes active. A straightforwad approach is random initial estimation. However, the number of P2P clients may change significantly, especially in flash crowd period; thus, a random estimation is unlikely to well approximate it. Furthermore, an under-estimation will result in capacity over-utilization; consequently, the division factors will be halved, leading to inefficiency in P2P capacity sharing in the following rounds.

We can take an ISP-aided initial estimation. Specifically, the ISP provides $\tilde{K}_i = \lceil \frac{p-\tilde{x}_i}{p-v_{i-1}} \rceil$ as a reference initial estimation. A new P2P client should obtain this reference and share the P2P capacity only when the network is under-utilized. This approach is desirable in that as Proposition 2 shows, under some mild assumptions, a single new P2P can join the network without causing the network to be over-utilized.

Proposition 2. *Suppose that the assumptions in Proposition 1 hold. If a single new P2P client joins a network and use* $\tilde{K}_i = \lceil \frac{p-\tilde{x}_i}{p-v_{i-1}} \rceil$ *in its initial estimation of division factor, then the network is under-utilized in the next interval.*

We next consider the implications for Internet charging models. One potential concern of the proposed ISP-compliant P2P design is that the upstream providers may thus have incentives to change their charging models. We believe that this is unlikely to happen. ISPs' traffic is still likely to have diurnal patterns, even with a large fraction of P2P traffic (see, *e.g.*, [10]) Consequently, an ISP's network should still be designed to handle the aggregated traffic peaks, regardless of P2P traffic. Thus, even if the traffic rate is flatten out at the targeted charging volumes (typically much lower than the peak rates) due to ISP-compliant P2P traffic, the ISP should still be able to handle all traffic.

6 Related Work

P2P traffic shaping (*e.g.*, [6,9]) and caching (*e.g.*, [3,8]) can be used to reduce bandwidth consumption of P2P applications. However, P2P applications often

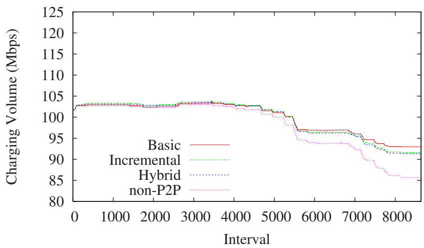

Fig. 7. Robustness of traffic volume prediction algorithms

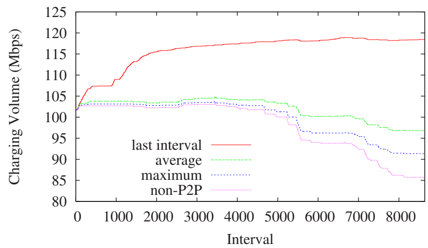

Fig. 8. Robustness of charging volume prediction algorithms

differ in control messages, and hide themselves using encryption. Many ISPs are reluctant to get involved in P2P content distribution by deployment of P2P caching devices. It remains unclear whether in the long run they can effectively reduce operational costs.

More recently, researchers have studied P2P network connectivity control to reduce P2P bandwidth consumption. In particular, Xie *et al.* proposed iTracker as an ISP portal service for disseminating network status and policy [11], which resulted in the formation of a Distributed Computing Industry Association (DCIA) [4] working group P4PWG and large-scale pilot studies on P4P. Aggarwal *et al.* proposed an oracle as an ISP service to rank the peers for P2P clients [2]. However, there lacks a study and it remains unclear in [2] how effectively ISPs can implement their policies by simply ranking the peers. Our framework differs from both approaches in that it imposes finer-grain control and has the potential benefit of implementing ISP policies more effectively.

7 Conclusion

In this paper, we present a new framework for ISP-compliant, peer-friendly P2P design. The framework enables ISP network to determine its capacity allocatable to P2P applications. We show that in typical percentile networks, an ISP can allocate a large amount of bandwidth to P2P applications without increasing its financial cost. We also show that P2P clients can use a distributed, peer-friendly algorithm to fairly share the allocated P2P bandwidth.

References

1. Message Stream Encryption, `http://www.azureuswiki.com`
2. Aggarwal, V., Feldmann, A., Scheideler, C.: Can ISPs and P2P systems cooperate for improved performance? In: ACM CCR (July 2007)
3. CacheLogic. Serving cached content, `http://www.cachelogic.com/products/cachepliance.php`
4. Distributed Computing Industry Association, `http://www.dcia.info`

5. Odlyzko, A.: Internet pricing and the history of communications. Computer Networks 36, 493–517 (2001)
6. Packeteer. Packeteer PacketShaper, `http://www.packeteer.com/products/packetshaper`
7. Parker, A.: The true picture of peer-to-peer filesharing (July 2004), `http://www.cachelogic.com`
8. Saleh, O., Hefeeda, M.: Modeling and caching of peer-to-peer traffic. In: Proceedings of ICNP 2006, Washington, DC (November 2006)
9. Sandvine. Intelligent broadband network management, `http://www.sandvine.com`
10. Saroiu, S., Gummadi, K.P., Dunn, R.J., Gribble, S.D., Levy, H.M.: An analysis of internet content delivery systems. In: Proceedings of OSDI 2002, Boston, MA (December 2002)
11. Xie, H., Krishnamurthy, A., Yang, Y. R., and Silberschatz, A. P4P: Proactive provider participation for P2P. Tech. Rep. YALEU/DCS/TR-1377, Yale University (March 2007)
12. Xie, H., Yang, Y.R.: A measurement-based study of the skype peer-to-peer VoIP performance. In: Proceedings of IPTPS 2007, Bellevue, WA (February 2007)