Caching the P2P traffic in ISP network

Mingjiang Ye, Jianping Wu, Ke Xu
Department of Computer Science and Technology
Tsinghua University
Beijing, P.R.China, 100084
yemingjiang@csnet1.cs.tsinghua.edu.cn, jianping@cernet.edu.cn, xuke@csnet1.cs.tsinghua.edu.cn

***Abstract*—The rise in peer-to-peer(P2P) networking has been tremendous in last several years. P2P traffic has significant impact on ISPs as it accounts for more than half of all traffic. Although many methods have been proposed to manage P2P traffic, little effort was spent on the deployment issues. In this paper, we have studied how to deploy P2P traffic cache devices in the backbone network to maximize the benefit of the ISP. A novel model is proposed to evaluate the benefits of deploying cache devices on different links. Guided by our model, two algorithms were developed to instruct the deployment of cache devices in the network. The experiment shows that deploying cached devices on less than 10% links can efficiently reduce the heavy load of the backbone network.**

Keywords- network management; peer-to-peer networks; traffic caching;

I. INTRODUCTION

Peer-to-peer (P2P) systems have gained tremendous popularity in the past few years. There have been numerous studies showing that traffic generated by P2P systems accounts for a major fraction of the Internet traffic [1-4]. Some studies have found that P2P traffic makes up to even 80-90 percent of local network traffic and 40-60 percent of backbone traffic, and keeps increasing [2-3]. The large volume of P2P traffic has negative impact on ISP including: (1) significantly increased load on the Internet backbone, hence, higher chances of congestion; (2) ISPs have to increase the investment on the capability of their backbones to satisfy the growing need.

Many researches [4-9] have been done on P2P traffic management, such as rate limiting the P2P traffic, using the local-aware P2P client, superpeering inside ISP network, caching the P2P traffic etc. Caching the P2P traffic could be very effective. Many researchers have been working on this area [7-9], such as analysing the impact of P2P traffic characteristics on caching and developing P2P traffic cache algorithm etc. Most work focuses on caching the P2P traffic in single link, but little effort has been spent on how to deploy the cache devices in the backbone network to maximize the benefit of the ISP.

In this paper, we investigate the integrated solution of deploying the P2P traffic cache devices in the network. Specifically, our contributions can be summarized as follows. First, we develop a novel model to evaluate the benefits of deploying P2P traffic management devices on different links, and propose link benefit utility function in the model, which can be used to model the value of the traffic on different links. Second, we illustrated how to evaluate the impact of P2P cache devices on the P2P traffic in the network. It is not simple since the effect of deploying a new device depends on the current deployment. Finally, guided by our model, two algorithms are proposed to specify how to deploy cache devices in the network.

The rest of this paper is organized as follows. In section 2, we summarize the related work. Section 3 describes the model. Some detailed issues of the model are illustrated in section 4. Section 5 describes the algorithms which solve the problems proposed in the paper. In section 6, we present the results of simulation. Section 7 concludes the paper.

II. RELATED WORK

Besides the user-unfriendly method, several P2P traffic management policies have been proposed which will not hurt the user experience.

The authors of [4] found that current P2P protocols are not ISP-friendly, since a large portion of existing local content was found to be downloaded from external peers. The authors suggest making P2P protocols locality-aware. But it is not easy for ISPs to change the P2P protocols used by their subscribers.

Controlling the super peers in P2P networks has been studied in [5-6]. The authors of [5] found that traffic localization using a peer selection policy at super peers can restrict P2P traffic to the local metro network as much as 40%. The authors of [6] further suggest these super peers provide additional features and services to encourage users to use these super peers. But the users may avoid using the super peers provided by ISP, because they may be afraid of operators seeing the files they are requesting.

Caching could be very effective in large P2P networks. This issue has been studied e.g. in [7-9]. The importance and feasibility of caching P2P traffic have been shown in these works. In [7], authors found that even for very small user populations, caching would save 40-60 % of the bandwidth. The caching algorithms for P2P traffic studied in [8-9] further showed that high byte-hit rates can be achieved for P2P traffic using relatively small space.

In this paper, we focus on deploying of P2P cache devices in ISPs' networks to maximize the income of their investment.

This research is supported by NSFC-RGC Joint Research Project (20731160014), NSFC Research Project (60473082), and 973 Project of China (2003CB314801).

III. PROBLEM AND MODEL

A. Problems

Caching P2P traffic can significantly save the bandwidth in the network. The objectives of caching schemes for web caching and video streaming are to minimize average start-up delays and to enhance stream quality. In contrast, P2P traffic caching is aimed to reduce traffic in the network. The paper focus on caching P2P traffic, but the model proposed is also suitable in caching other types of the traffic.

Caching algorithms for P2P traffic on one link has been studied in a lot of literatures, but there is little work on how to deploy P2P traffic cache devices in the network, i.e., on which links should the devices be deployed? The problem is not as simple as it first looks. The policy of deploying on the most congested links may fail to find the optimal solution, because the benefit from deploying a cache device may be impaired by other deployed ones.

To illustrate the problem, we simply assume a cache device can reduce 50% P2P traffic on a link. In Fig. 1, the P2P traffic from N2 to N1 is 800Mb/s, while the P2P traffic from N2 to N3 is 100Mb/s. In the simple case of deploying only one P2P cache device, the intuitionist policy of picking the most congested link will choose link3. But we can find that deploying on link1 is better than deploying on link3 according to Table 1, because deploying a cache device on link1 can help link1 and link3, while deploying on link3 only helps itself.

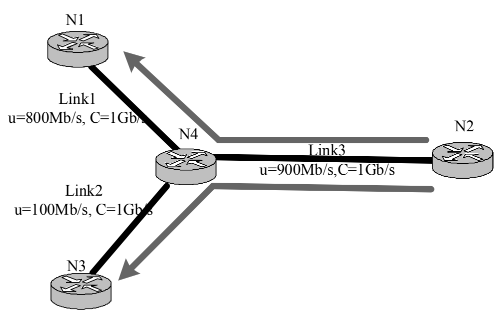

Figure 1. Sub-optimality network performance when using the policy of picking the most congested link

TABLE I. BENEFIT OF DEPLOYING ONE CACHE DEVICE

Deploy ment Schema	**Reduced Traffic(Mb/s)**			
	Link1	***Link2***	***Link3***	***Total***
Link1	400	0	400	800
Link2	0	50	50	100
Link3	0	0	450	450

B. Modelling the benefit

The ISP's network is considered as a graph in the model. Suppose there are n nodes in the network graph, denoted by $N_i (1 \le i \le n)$, and m direct links, denoted by $l_i (1 \le i \le m)$. A physical link is considered as two direct links in the graph. The traffic volume and the bandwidth of link l_i are denoted by u_i and C_i.

The benefit utility function of link l_i is defined as $\psi_i(\eta_i)$ (η_i is link utilization), which represents the benefit of reducing a small unit of traffic from link l_i. When a small unit of traffic dv is reduced from link l_i, the benefit gained from the traffic is denoted as:

$$B(dv) = \psi_i(\eta_i) dv . \tag{1}$$

If total amount of v_i traffic is reduced from the link l_i, the total benefit of the link l_i is:

$$B_i = \int_0^{v_i} \psi_i(\eta_i) dv . \tag{2}$$

When the traffic on link l_i is reduced by some amount v, the link utilization of link l_i is changed to $\eta_i = \frac{u_i - v}{C_i}$. So the benefit of link l_i is

$$B_i = \int_0^{v_i} \psi_i(\eta_i) dv = \int_0^{v_i} \psi_i\left(\frac{u_i - v}{C_i}\right) dv . \tag{3}$$

The total benefit of the network is calculated by summing up the benefit of each link, as in (4).

$$B_{total} = \sum_{i=1}^{m} B_i = \sum_{i=1}^{m} \int_0^{v_i} \psi_i\left(\frac{u_i - v}{C_i}\right) dv . \tag{4}$$

To calculate the benefit of a deployment plan using (4), the benefit utility function $\psi_i(\eta_i)$ has to be defined and the reduced traffic in each link should be figure out. These two issues are discussed in section 4.

IV. ISSUES IN MODELLING

A. Link Benefit Utility Function

The values of per-unit traffic on each link are different. For example, reducing traffic on a heavy congest link is obviously more important than reducing traffic on a light load link. ISP may also pay more attention to the traffic on important links. Link benefit utility function is used to give different weight to links according to their link utilization and importance.

Several typical benefit utility functions can be adopted. Constant functions mean the benefit of the reduced traffic only depends on the traffic volume on the links. Linear functions mean the benefit gained from reducing a unit of traffic is linear decreasing with the link utility. Other forms of functions, such as sub-linear functions and super-linear functions, are also possible candidates. Linear functions are used in our simulation, but the effect of different benefit utility functions requires further study.

B. Reduced Traffic

Using the model in section 3, the benefit of deployment can be calculated if the traffic removed in each link is known. In this section, we illustrate how to evaluate the effect of deploying P2P cache devices in a network.

A P2P traffic matrix is used to represent the P2P traffic between all pairs of user nodes in the network graph. User nodes are networks of other ISPs or access networks, not intermediate routers. The P2P traffic originates from these nodes and ends at these nodes in the network. a_{ij} is used to denotes the volume of P2P traffic which is originated from node N_i and ends at node N_j . The path from user node N_i to user node N_j is defined as :

$$P_{ij} = \{ l_{i1}, l_{i2}, l_{i3} .. \ l_{ik} \}. \tag{5}$$

When a P2P cache device deploying on link l_x, the path P_{ij} is split into two paths $P_{ij-front}$ and P_{ij-end} by link l_x.

$$P_{ij-front}^{l_x} = \{ l_{i1}, l_{i2}, \ l_{i3} \dots \ l_{i(s-1)} \} \ (l_x = l_{i(s-1)}),$$
$$P_{ij-end}^{l_x} = \{ l_{is}, l_{is+1} \dots \ l_{ik} \}. \tag{6}$$

Since P2P traffic is cached and served by P2P cache device on link l_x, the traffic on links of path $P_{ij-front}^{l_x}$ is reduced while the traffic on links of path $P_{ij-end}^{l_x}$ is not affected.

Suppose that the P2P cache devices are deployed on links in link set Ω ($\Omega = \{l_{x1}, l_{x2}, \dots \ l_{xt}\}$), the reduced traffic on link l_y is:

$$v_{l_y} = \sum_{\{i,j | l_y \in P_{ij-front}^{l_z}, l_z \in \Omega\}} v_{ij} = \sum_{\{i,j | l_y \in P_{ij-front}^{l_z}, l_z \in \Omega\}} \lambda a_{ij} \tag{7}$$

λ is the byte hit ratio which represents the fraction of total P2P traffic served by a cache server. v_{l_y} is calculated by summing up of the cached traffic between the affected node pairs. These pairs satisfy two conditions: (1) the path between the node pair contains link l_y , (2) there is a link $l_z \in \Omega$, and $l_y \in p_{ij-front}^{l_z}$. Replacing (7) into (4), we can calculate total benefit of the deployment schema.

V. ALGORITHMS

The complexity of the problem is analysed in this section, and the heuristic and exact algorithms are proposed to solve the problem.

A. Complexity Analysis

The cache deployment problem can be written as: Given topology of the network with m links, each link with a deployment cost, determine which links to deploy cache devices so that the total cost is less than some given cost and the total benefit is as large as possible.

We use $C = (c_i)$ to represent the deployment cost of each link and c is the limitation of the cost. When $c_i = 1, \forall i$, the problem is equal to finding the best deployment schema, with the limitation of the number of the links to deploy the cache devices. The deployment schema $X = (x_i)$ is a vector, where the variable x_i =1 if the link l_i is selected to deploy the cache devices and x_i =0 otherwise.

The problem can be described as an optimization problem:

$$\text{Max: } B_{total} = \sum_{i=1}^{m} B_i(X)$$
$$\text{s.t.: } \sum_{i=1}^{m} x_i c_i \le c \tag{8}$$

Note that $B_i(X)$ is the benefit of the link l_i in deployment schema X. The benefit of each link is not constant, which is depended on the deployment schema, so the problem is not an equivalent of 0/1 Knapsack Problem.

Lemma 1: The cache deployment problem can't be solved in polynomial time.

Proof: Considering a full-connected network, which is a special case of the problem. Since the path between any pairs of nodes is a separate single link, traffic on a link is only affected by the cache device on it.

We defined the deployment schema X_i^* = (0, 0, ..., x_i =1, ... ,0,0) which means there are no cache devices deployed in the network except link l_i.

In a full-connected network, the traffic on link l_i is only affected by the device deployed on link l_i. We have:

$$B_i(X_i) = \begin{cases} B_i(X_i^*), x_i = 1 \\ 0 \quad , x_i = 0 \end{cases} \quad \forall X_i \in X \tag{9}$$

The problem can be rewritten as

$$\text{Max: } B_{total} = \sum_{i=1}^{m} x_i B_i(X_i^*)$$
$$\text{s.t.: } \sum_{i=1}^{m} x_i c_i \le c \ , \ x_i = 0 \text{ or } 1 \tag{10}$$

This problem is equal to the 0/1 knapsack problem. Given a container of capacity c, and m items, each having profit $B_i(X_i^*)$ and cost x_i, k=1, ..., m.

As proved, 0/1 knapsack problem, which is NP-Complete, is polynomial-time reducible to the cache deployment problem, so this problem can't be solved in polynomial time.

B. Greedy Algorithm

For large scale networks, we proposed a heuristic algorithm to solve the problem. Many heuristic methods have been proposed for 0/1 knapsack problem. But as mentioned in last section, the problem is not an equivalent.

```
Greedy Algorithm
DeploySet ={};
LeftCosts = c;
CandidateLinks = {all links of network}
While (true){
   maxMargin =0;  t = -1;
   for (int i= 1; i< m;  i++){
      if ( c_i  >  LeftCosts)
         CandidateLinks = CandidateLinks– { l_i }
      if ( l_i ∉ CandidateLinks) continue;
      m=(B(DeploySet ∪ { l_i }) - B(DeploySet))/ c_i
      if (maxMargin  <  m ){
         t = i; maxMargin  =  m ;}
   }
   if (t == -1)
      break;
   DeploySet = DeploySet  ∪  { l_t }
   CandidateLinks = CandidateLinks– { l_t }
   LeftCosts = LeftCosts– c_t ;
}
return DeploySet;
```

In greedy algorithm, the link with maximum benefit per cost is selected in each round until no candidate link left. The benefit of candidate links must be recalculated every round since it depends on the current deployment solution. To calculate the benefit of a candidate link after deployed, We subtract the current benefit of deployment link set(DeploySet) from the benefit of set(DeploySet $\cup$ { l_i }).

C. Branch and Bound Algorithm

Although the problem can't be solved in polynomial time, exact methods can still be used to find the optimal solution for small and medium scale topologies, we proposed a branch and bound algorithm to find the optimal result.

In the pseudo code of the algorithm, L is used to represent a deployment link set. $c(L)$ is the cost of L, and $B(L)$ is the deployment benefit of L. $G(L)$ is defined as the maximize benefit which can be achieved at last, under the limitation that the final deployment link set must contain L.

```
Branch and Bound Algorithm
DeploySet = output of Greedy Algorithm;
LowerBound = B(DeploySet)
H is a max-heap whose elements are deployment link sets and compared with their upper bound benefit.
add empty link set φ to H
While (true){
  L = get top element from H
  if ( U(L)  <= LowerBound)
        break;
  for (int i= 1; i< m, i++){
     if ( l_i ∉ L && ( c(L) + c(l_i) <= c){
        L_new = L ∪ {l_i}
        if ( B(L_new) >LowerBound){
           DeploySet = L_new ;
           LowerBound = B(L_new)
        }
        U(L_new)  = evaluate( L_new , c)
        if ( U(L_new) >LowerBound)
           add L_new to H
  }}}
return DeploySet;
```

The algorithm searches all the possible candidates, and uses the lower bound and upper bound to limit the search space. The initial lower bound is calculated by using greedy algorithm. To limit the search space, $U(L)$, the upper bound of $G(L)$, is evaluated as following algorithm.

```
evaluate (L, c)
LinkArray is a pre-constructed array of all links in network and sorted in descending order by B({l}) / c({l}) of each link.
cost = c(L) ;
benefit = B(L)
for (int i= 1; i< m, i++){
   l = LinkArray [i];
   if ( l ∉  L ){
      benefit = benefit + B({l})
      cost = cost +  c({l}) }
   if ( cost > c)  break;
}
return benefit;
```

VI. Evaluation

A simulator is written in Java for the evaluation. The simulation is run on a PC with Pentium 4 CPU of 2.66G Hz and 1GBbyte RAM.

A. Experiment Setup

Topologies of our campus network, which has tens of thousands of Ethernet users, are used in this simulation to study the effect of our model. The bandwidth of each link is 1Gb/s. The link utilization of each link varies from 1% to 94%. The load of some links is very heavy, while some are quite light. The local ISP network is more complex, which has more than fifty 1Gb/s links and several 10Gb/s links. The utilization of each link varies from 20% to 70%.

The P2P Traffic Matrix is generated randomly with the constraint that most P2P traffic exchanges with external peers. The benefit utility function of the internal link is set to 2000η Mb/s. The link utility function of the peer link is 2000η Mb/s, and the benefit utility function of the upstream link is 2000 Mb/s. The byte hit ratio λ is 0.5 in the simulation. The cost of each link is set to one unit, so the limitation of the cost can be expressed as the number of cache devices to be deployed in the network.

B. Simulation Result

We have compared three algorithms in our simulation. One is the link utilization based algorithm, which picks up the most congested links, the other two are greedy algorithm and branch and cut algorithm proposed in the paper.

The total benefit is shown in Fig. 2, and the average link utilization of the network is shown in Fig. 3. Limited by the computation power, the branch and cut algorithm can only be used to find the optimal result when the number of the cache devices is less than 10. In these cases, the results of the branch and cut algorithm and the greedy algorithm are exactly the same.

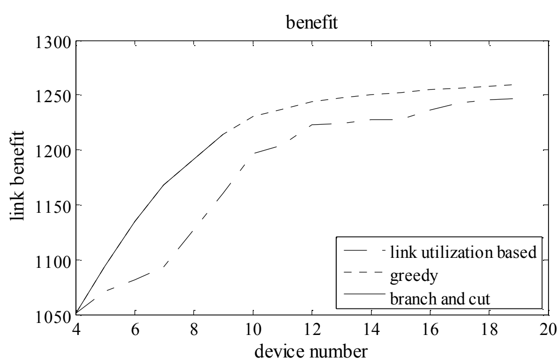

Figure 2. Total benefit achieved by each algorithm

When deploying less than 5 cache devices, the results of the link utilization based algorithm and the greedy algorithm are the same. But when deploying more cache devices, the greedy algorithm performances much better than the simple link utilization based algorithm. It can achieve more gain from the deployment and also lower average link utilization of the network. Deploying 10 cache devices, which is about 10% of the number of links in network, the link utilization improvements by greedy algorithm is significant.

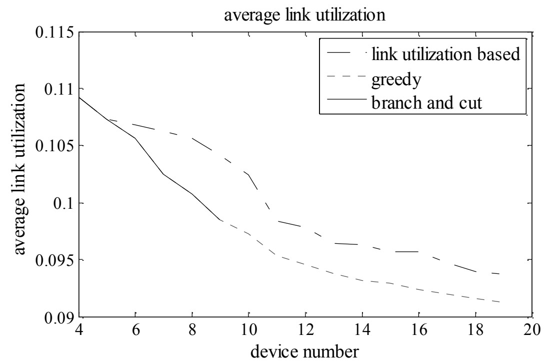

Figure 3. Average link utilization

VII. CONCLUSION

In this paper, we proposed a novel model to evaluate the benefit of deploying the P2P traffic management devices in ISPs' networks, giving different weight to links according to their link utilization and importance. According to the model, we proposed two algorithms to instruct deployment of P2P cache devices in ISP networks to maximize the income of their investment.

We are currently cooperating with a local ISP on managing the P2P traffic in its network by using our method. In the future, we intend to investigate the combination of using the cache devices and traffic redirection method in application layer to maximize the benefit from the investment on the cache devices.

REFERENCES

[1] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks,". IEEE/ACM Transactions on Networking, vol. 12, no. 2, pp. 219–232, 2004.

[2] L. Plissonneau, J.L. Costeux, P. Brown, "Analysis of. Peer-to-Peer Traffic on ADSL", Passive and Active. Measurement 2005, Boston (MA) USA, pp. 69–82, March 2005.

[3] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport Layer Identification of P2P Traffic", Proceedings. of the 4th ACM SIGCOMM conference on Internet measurement, pp. 121–134, 2004.

[4] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should internet service providers fear peer-assisted content distribution?" Proceedings. of the 5th ACM SIGCOMM conference on Internet measurement, pp. 63–76, 2005.

[5] S. Singh, S. Ramabhadran, F. Baboescu, and A. Snoeren, "The case for service provider deployment of super-peers in peer-to-peer networks", Workshop on Economics of Peer-to-Peer Systems, Berkeley, 2003.

[6] T. Hamada, K. Chujo, T. Chujo, and X. Yang, "Peer-to-peer traffic in metro networks analysis, modeling, and policies", Network Operations and Management Symposium (NOMS), pp. 425–438, 2004.

[7] K. Gummadi , R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload," Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOS'03), 2003.

[8] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Wozniak, "Cache replacement policies revisited: The case of P2P traffic", in Proc. of the 4th International Workshop on Global and Peer-to-Peer Computing (GP2P'04), pp. 182–189, 2004.

[9] O. Saleh and M. Hefeeda, "Modeling and caching of peer-to-peer traffic", In Proc. of 14th IEEE International Conference on Network Protocols (ICNP'06), pp. 249–258, 2006.