Optimal Neighbor Selection in BitTorrent-like Peer-to-Peer Networks

Hao Zhang
University of California, Berkeley
Berkeley, CA 94720, USA
zhanghao@berkeley.edu

Ziyu Shao
The Chinese Univ. of Hong Kong
Shatin, NT, Hong Kong
zyshao@ie.cuhk.edu.hk

Minghua Chen
The Chinese Univ. of Hong Kong
Shatin, NT, Hong Kong
minghua@ie.cuhk.edu.hk

Kannan Ramchandran
University of California, Berkeley
Berkeley, CA 94720, USA
kannanr@eecs.berkeley.edu

ABSTRACT

We study the problem of neighbor selection in BitTorrent-like peer-to-peer (P2P) systems, and propose a "soft-worst-neighbor-choking" algorithm that is provably optimal. In practical P2P systems, peers often keep a large set of potential neighbors, but only simultaneously upload/download to/from a small subset of them, which we call *active* neighbors, to avoid excessive connection overhead. A natural question to ask is: which active neighbor set should each peer choose to maximize the global system performance? The combinatorial nature of the problem makes it especially challenging. In this paper, we formulate an optimization problem and derive a distributed algorithm. We remark that our solution has a similar favor compared to the *worst neighbor choking* and *optimistic unchoking* neighbor selection algorithms that are implemented by BitTorrent. However, it encourages peers to stick to better performing neighbors for longer time and is provably globally optimal. Our proposed solution is easy to implement: each peer periodically waits for a constant period of time that depends on the size of the potential neighbor set and the aggregated utility of the active neighbors, *chokes* (drops) one of its current active neighbors with probability proportional to an exponential weight on the utility of the corresponding link, and randomly *unchokes* (adds) a new neighbor from its potential neighbor set. Our theoretical findings provide insightful guidelines to designing practical P2P systems. Simulation results corroborate our proposed solution.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications

General Terms

Algorithms, theory

1. INTRODUCTION

Consider a P2P overlay network represented by a directed graph $G = (V, E)$, where V denotes the set of all the nodes and E is the set of all the *upload* links. Assume that each node v has a certain upload link capacity $C_v \geq 0$ and has no limit on the download link capacity. Each node v has a potential *upload* neighbor set, denoted by N_v^p, which it can choose to upload to. However, each node v can upload to at most B_v neighbors simultaneously. We call this constraint an upload connection degree bound B_v. We refer to a specific peer neighboring connections a *topology configuration*, denoted by f. A configuration is essentially a snapshot of the current *active* P2P connection overlay graph. Let N_v^f be the set of neighbors that node v is currently *uploading* to under configuration f. Denote by $\mathcal{F}$ the set of all possible topology configurations in which the active neighbor set at every node satisfies its corresponding connection degree bound. Table 1 lists the relevant notation. Our goal is to maximize the overall utility *jointly* over peers' upload bandwidth allocation and peer neighbor selection in a distributed way. We formulate the problem as follows:

Table 1: Key Notations

Notation	Definition
$\mathcal{F}$	the set of all peer neighboring configurations
V	the set of all peers
N_v^p	peer v's potential upload neighbor set
N_v^f	peer v's active upload neighbor set under f
x_{vu}^f	upload rate from peer v to peer u under f
C_v	upload capacity of peer v
B_v	outgoing upload connection bound of node v
U_v	concave utility function of node v
g_f	system utility under f

Note: we use bold-type to denote vectors.

$$\begin{aligned} \max_{f\in\mathcal{F},\boldsymbol{x}^f} \quad & \sum_{v\in V} U_v(\boldsymbol{x}_v^f) && (1) \\ \text{s.t.} \quad & \boldsymbol{x}_v^f = \{x_{uv} \mid v \in N_u^f, \forall u \in V\},\ \forall v \in V \\ & \sum_{u\in N_v^f} x_{vu}^f \leq C_v,\ \forall v \in V \\ & |N_v^f| \leq B_v,\ \forall v \in V \end{aligned}$$

This is a mix convex-combinatorial problem. Adapting Lagrange dual decomposition and Markov approximation techniques [1], we propose to solve it by letting each peer v running a distributed algorithm, stated in Algorithm 1, independently. where $h_u(x_{vu}^f) =$

SIGMETRICS'11, June 7–11, 2011, San Jose, California, USA.
ACM 978-1-4503-0262-3/11/06.

Algorithm 1 Rate Allocation and Neighbor Selection Algorithm

1: **Initialization:** Set $x_{vu} = 0$, $\lambda_v = 0$ and $t = T_v$ where T_v is the count-down time. Iterate:
2: Receive from all the active neighbors $u \in N_v^f$ their marginal utility value $h_u(x_{vu}^f)$ at the current upload rate x_{vu}^f, and then perform the following updates.
3: $x_{vu} \leftarrow x_{vu} + \varepsilon(h_u(x_{vu}^f) - \lambda_v)_{x_{vu}^f}^{[0,+\infty)}$, $\forall u \in N_v^f$
4: $\lambda_v \leftarrow \lambda_v + \delta(\sum_{u \in N_v^f} x_{vu} - C_v)_{\lambda_v}^{[0,+\infty)}$
5: Allocate and sends packets to the active neighbors according to the new rates x_{uv}.
6: **if** $t = 0$ **then**
7: Choke neighbor u with probability $\frac{\exp(-\beta x_{vu}^f h_u(x_{vu}^f))}{\sum_{u' \in N_v^f} \exp(-\beta x_{vu'}^f h_{u'}(x_{vu'}^f))}$, randomly unchoke a new neighbor from the inactive potential set to replace u, and set $x_{vu} = 0$ and $t = T_v$.
8: **end if**
9: $t \leftarrow t - 1$.

$\frac{\partial U_u(\boldsymbol{x}_u^f)}{\partial x_{vu}^f}$ is the *marginal utility* of node u with respect to its download link rate x_{vu}^f, $\varepsilon, \delta > 0$ are small constants.

2. MAIN RESULTS

THEOREM 1. *If the count down time T_v in Algorithm 1 at node v is exponentially distributed with mean*

$$\frac{1}{\tau\left(|N_v^p| - |N_v^f|\right)\sum_{u' \in N_v^f} \exp(-\beta x_{vu'}^f h_{u'}(x_{vu'}^f))} \quad (2)$$

then the overall system utility $g = \sum_{v \in V} U_v(\boldsymbol{x}_v^f) \to \bar{g}$ as $\beta \to \infty$, and:

$$|g_o - \bar{g}| \leq \max_{v \in \mathcal{V}} U_v(C_v) \quad (3)$$

where g_o is the optimal solution to problem (1).

THEOREM 2. *The average performance $\bar{g}$ is insensitive to the distribution of the count-down time $T_v, v \in V$ as long as the mean of the count down time satisfies (2).*

We omit the proof details due to the space limit. We make the following remarks.

The proposed solution is fully distributed, i.e., each peer runs the rate allocation and neighbor selection algorithm independently. The optimality gap $\max_{v \in \mathcal{V}} U_v(C_v)$ is quite small when the total number of nodes in the system $|\mathcal{V}|$ is large.

The neighbor selection algorithm is also intuitive: (a) the larger the inactive potential set $|N_v^p| - |N_v^f|$, the shorter time a peer should wait till he finds a new peer to upload to; (b) the better the overall marginal aggregated utility $\sum_{u' \in N_v^f} \exp(-\beta x_{vu'}^f h_{u'}(x_{vu'}^f))$ to the neighbors, the longer time a peer should wait before finding new neighbors; and (c) the larger the aggregated marginal utility (thus the smaller $\exp(-\beta x_{vu}^f h_u(x_{vu}^f))$), the less likely the corresponding active neighbor will be choked and vice versa. This intuitive strategy encourages peers to stay longer in better performing configurations. We call our neighbor selection algorithm the "*soft-worst-neighbor-choking*" algorithm.

Surprisingly, the heuristic approaches of "tit-for-tat" choking and "optimistic unchoking" that are implemented BitTorrent [2] is a similar version to our "*soft-worst-neighbor-choking*" algorithm. In BitTorrent, a peer periodically chokes its upload to an active peer with the *worst* download rate with a fixed period of 10 seconds. In our case, a peer chokes neighbors with the probability proportional to an exponential weight on the aggregated marginal utility of the neighbors. Since the probabilities are exponentially weighted, a peer essentially "softly" chokes an neighbor with worst aggregated marginal utility. Also in BitTorrent, a peer "optimistically" unchokes a *random* new inactive neighbor periodically every 30 seconds, in order to explore new peers with potentially better download rates. In our algorithm, peers also randomly finds new neighbors from time to time. However, our algorithm waits for a longer time when the system performance is at a better state and vice versa. This helps drive the system to move faster to and stay longer in better configurations.

Our proposed algorithm is generalizable to other P2P systems. The same peer selection algorithm can be distributively implemented, and only a different utility function needs to be plugged in for different applications.

Table 2: Peer upload capacity distribution

Upload (kbps)	512	640	768	896	1024	1152	1280
Fraction (%)	5	10	5	40	15	10	15

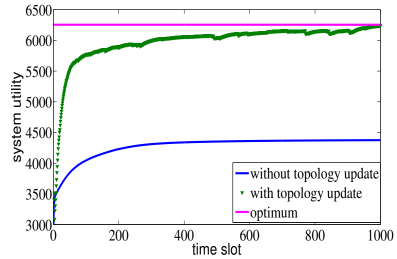

Figure 1: System utility with and without topology building.

3. EXPERIMENTS

3.1 Setup

We set the number of peers $|V| = 100$. Peers have upload capacities draw from the distribution shown in Table 2. Each peer can have a potential neighbor set of $|N_v^p| = 20$, and a degree bound of $|B_v| = 2$.

We set the utility function of each user v as:

$$U_v(\boldsymbol{x}_v^f) = \begin{cases} |\boldsymbol{x}_v^f|_1 - \frac{|\boldsymbol{x}_v^f|_1^2}{2r} & \text{if } |\boldsymbol{x}_v^f|_1 \leq r \\ \frac{r}{2} & \text{if } |\boldsymbol{x}_v^f|_1 > r \end{cases}$$

where $|\boldsymbol{x}_v^f|_1 = \sum_{u: v \in N_u^f} x_{uv}^f$ is the summation of the received rate from its download neighbors, and $r = 1024$kbps. We also set the step sizes $\varepsilon = 2$ and $\delta = 0.4$ for the bandwidth allocation algorithm, and set $\beta = 5$ and $\tau = 1$ in the topology update algorithm. We can see that the proposed algorithm performs quite well. The proposed solution has a significant gain compared to a static topology setting and is close to the optimum.

4. REFERENCES

[1] M. Chen, S. C. Liew, Z. Shao, and C. Kai. Markov approximation for combinatorial network optimization. In *Proc. of IEEE INFOCOM*, San Diego, CA, USA, 2010.

[2] B. Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, volume 6. Berkeley, CA, USA, 2003.