Fang ZUO[1], Wei ZHANG[2]

East China Normal University (1), East China Normal University (2)

A game-theoretic approach for overcoming selfish routing in P2P networks

***Abstract**. We introduce an evolution game based model to study the temporal behaviors of selfish nodes in P2P networks. So far, most of the analysis of selfish routing is concerned with static properties of equilibria which is one of the most fundamental paradigms in classical Game Theory. By adopting a generalized approach of evolutionary game theory, we extend the model of selfish routing to study the dynamical behaviors of nodes. Also give an algorithm and experiment values on how to improve P2P traffic efficiency by evolutionary game model.*

Streszczenie:** W opracowaniu, do badań zachowań w czasie samolubnych węzłów w sieciach P2P, wprowadzono model oparty o grę ewolucyjną. Dotychczas większość analiz samolubnego trasowania koncentruje się na statycznych własnościach równowagi, co jest najbardziej fundamentalnym paradygmatem w klasycznej teorii gier. Przez przyjęcie uogólnionego przybliżenia ewolucyjnej teorii gier rozszerzono model samolubnego trasowania na badania dynamicznych zachowań węzłów. Podano zarówno algorytm jak i przykłady eksperymentalne polepszenia trasowania P2P przez zastosowanie ewolucyjnego modelu gier.* ***Zastosowanie równań teorii gier do opanowania samolubnego trasowania w sieciach P2P

Keywords: Selfish routing, P2P networks, Imitative mechanism; Evolution game.
Słowa kluczowe: Trasowanie samolubne, Sieci P2P, Gra ewolucyjna

Introduction

The emergence of peer-to-peer (P2P) is a popular and powerful networking paradigm. The basic idea of P2P is to organize a virtual overlay network on top of the physical network so that nodes in the overlay can be customized to corporate in a flexible pair-matching manner without modifying native routers. However, each peer in P2P networks usually routes its traffic on the minimum-latency path available to it without considering the network congestion and routing chaos, which leads routing and other network traffic problems for both P2P applications and ISP providers.

In response to this kind of issues, network scholars have considered multiple new P2P optimization techniques. Unfortunately, none of them appear to be fully satisfactory. For examples, some Internet service providers eliminate P2P traffic flow by using feature code matching method to identify P2P data packets [1,2], another proposal is to deploy P2P caching devices to cut down bandwidth consumed by P2P applications [3,4], and some other ways to solve the P2P application traffic optimization problem is to build distributed application-level services for location and path selection in order to enable peers to estimate their position in the network and to efficiently select their neighbours [5-7].

In this context, we firstly introduce a P2P network evolutionary game model as the P2P network efficient traffic optimization mechanism. In the absence of regulation by some central authority, we assume that each network peer routes its traffic on the minimum-latency path available to it. In such a selfishly motivated assignment of traffic to paths will not minimize the overall latency, this lack of regulation carries the cost of decreased network performance. Our goal is (1)to route traffic such that the overall average latency is minimized by using node evolutionary game mechanism, namely counteracting and imitating the other node's routing strategies; and (2)to steer the whole network system towards a stable state (also can be called equilibrium state, namely both peer optimal and network system optimal),where at such equilibrium no network user has an incentive to switch paths or route and this occurs when all traffic flow travels on the same minimum-latency paths.

The contribution of this work is (1)to propose a P2P overlay traffic routing model with evolutionary game-theoretic analysis to help understand how overlay traffic converge to an equilibrium state, doing so involves applying techniques from complexity theory to the problem of finding a game-theoretic equilibrium and P2P traffic routing features; (2)implement an algorithm based on such evolutionary game-theoretic analysis on how to overcome selfish routing and routing congestion problems in P2P network and show some positive results in our simulations.

P2P traffic routing model with evolution dynamics

(a). Preliminary

In P2P network, since the interactions of nodes are happened randomly, each node is lack of global network information and other peers' strategy information. This characteristic of nodes can be regarded as bounded rationality in the incomplete information peer game. Regarding to the different strategies, the adjustment period of a node selecting its dominant strategy is a gradual process, which can be model by imitative dynamics of evolution game to simulate nodes' learning and strategic adjustment process in network. In the following context, we will give some important definitions and preliminaries about our research.

Considering an initial population of peers in P2P network, each node is assigned a pure strategy. At some point of time, each node pairs another node chosen uniformly at random and observes its own and its opponent payoff and decides whether to imitate its opponent or not by adopting its strategy with probability proportional to the payoff difference. Nodes interaction is regarded as evolutionary game with a payoff function *F*. Let *A* denote the set of actions available to both players, and let $\Delta(A)$ denote the set of probability distributions or mixed strategies over A, then $F : \Delta(A) \times \Delta(A) = \Re$. Suppose that there is a (1−ε) fraction nodes who play strategy s, and call these nodes—incumbents, and suppose that there is a ε fraction who play t, and call these nodes—mutants. The strategy s is an ESS if the expected payoff of a node playing s is higher than that of a node playing t. Since an incumbent will meet another incumbent with probability (1−ε) and it will meet a mutant with probability ε, we can calculate the expected payoff of an incumbent, which is simply $(1-\varepsilon)F(s \mid s) + \varepsilon F(s \mid t)$. Similarly, the expected payoff of a mutant is $(1-\varepsilon)F(t \mid s) + \varepsilon F(t \mid t)$. Thus we come to the formal definition of the P2P-ESS according to reference [8,10].

Definition 1: If nodes interact, a strategy *s* is a P2P evolutionarily stable strategy (P2P-ESS) for the P2P evolutionary game given by payoff function F, if for every strategy t $\neq$ s, there exists a ε_t such that for all $0<\varepsilon<\varepsilon_t$, $(1-\varepsilon)F(s|s)+\varepsilon F(s|t)>(1-\varepsilon)F(t|s)+\varepsilon F(t|t)$.

Definition 2: Evolution game imitative dynamics in P2P. Let x_i denote the fraction of nodes playing strategy *i* and node population vector $\vec{x}$.

$$\dot{x}_i = \lambda(\vec{x})x_i[Ef\ (x_i)_{payoff} - \overline{E}\ f_{payoff}] = \lambda(\vec{x})x_i((A\vec{x})_i - \vec{x}A\vec{x}) \tag{1}$$

Where: $A=(a_{ij})$ is the payoff of strategy *i*. The function λ(x) accounts for the growth rate of node's population. As a fact, one can join or leave a P2P network freely and the growth rate of the node's population does not depend on the network's latency; so that we choose λ(x) = 1 or any other constant to replace λ(x) in following context.

Let G = (V, E) be a P2P overlay network with latency function $L(*)$ and with vertex set V, edge set E, and k source-destination vertex pairs { s_1, t_1} ,..., { s_k , t_k } . We also assume that from source vertex s to destination vertex t denote the set of s-t paths by P. Each edge e∈E is given a load-dependent latency function that we denote by $\ell_e(*)$. We assume that ℓe is nonnegative, continuous, and non-decreasing. The latency of a path P with respect to a P2P traffic flow f is then the sum of the latencies of the edges in the path, denoted by $\ell_P(f) = \sum_{e\in P} \ell_e(f_e)$. We call the triple (G, t, ℓ) a P2P network traffic model. With respect to a finite and positive traffic rate t, where $\sum_{p\in P} f_p = t$. We define the payoff C(f) of a flow f incurred by a node in G choosing path p, $C(f) = \sum_{p\in P} \ell_P(f) f_P = \sum_{e\in E} \ell_e(f_e) f_e$.

Definition 3: Flows at Nash Equilibrium. A flow f feasible for (G, t, ℓ) is said to be at Nash equilibrium (or is a Nash flow) if for every two s-t paths P1, P2 ∈P with $f_{P1} > 0$, $\ell_{P1}(f) \le \ell_{P2}(f)$. Nash flow means the latency of unused paths is equal or even greater than the latency of used paths. In particular, all latencies of used paths belonging to the same node group or the same node type are equal.

Definition 4: Equivalence of equilibrium and optimal flows. Optimal Flows means every flow that minimizes total latency. As we can see, the optimal flow is the P2P traffic system optimal solution; the situation in such optimal solution is often an ideal state, where most optimization methods can not achieve such a best situation.

Assuming mild extra conditions on the latency functions of an instance, there is a well-known characterization of optimal flows that mirrors the definition of Nash flows. Let (G, t, ℓ) have the property that, for each edge e, the function $x \cdot \ell_e(x)$ is convex and continuously differentiable. We define the marginal cost function $C(e)= x \cdot \ell_e(x)$. Then, a flow f' for (G, r, ℓ) is optimal if and only if it is at equilibrium flow for (G, r, f').

(b). P2P routing model with imitative dynamics

We use replicate dynamics mechanism to analyze the evolutionary trend of strategies among nodes. Peers adopt replication or imitation method for choosing and updating their strategies.

We are given a P2P network and a rate of traffic between a source peer node and a destination peer node, and seek an assignment of traffic to source-destination paths. We assume that each peer node controls a negligible fraction of the overall traffic, so that feasible assignments of traffic to paths in the network can be modeled as end to end flows in the triple (G, t, ℓ) . We also assume that the time needed to traverse a single link of the network is load-dependent, that is, the common latency by all traffic on the link increases as the link becomes more congested. In the absence of network regulation, peer nodes often act in a selfish manner. Under this assumption, our aim is to expect network traffic to converge to Nash flow.

Generally speaking, the latency functions are strictly increasing, and then Nash flows are ESS, the proof referred to [8,9]. Consider an initial population of peer nodes in which each node is assigned an arbitrary pure strategy (original arbitrary route from itself to its target node). At each point of time, each node plays against an opponent chosen uniformly at random from its neighborhood network domain. The node observes its own and its opponents latency as payoff and decides to imitate its opponent by adopting its strategy. One could argue that how often it is not possible to observe the opponent's latency. In this case, consider a random aspiration level for each node. Whenever a node falls short of this level, it adopts a regular observed strategy. We use evolution dynamics formulation to express the above scenario, the evolution dynamics process can be defined as:

$$\frac{dx}{dt} = \lambda(\vec{x})x[l(\vec{x}) - \bar{l}(\vec{x})] \tag{2}$$

To analyze the dynamics to either a Nash flow, it is necessary to compute the rate of change of the amount of flow over each path. we will use the notation x′to denote the derivative with respect to time of the variable x, that is, x′= dx/dt .

Let x_p the amount of P2P flow routed over path p , $x_e = \sum_{e\in P} x_p$ is the total load of edge e. And p can also be seen as the fraction population of nodes choosing route p at some given point of time, that is, x_p is a probability of nodes selecting route p.

We combine the individual values x_p into a vector X. The vector X=(x_1, x_2… x_P), which is indexed by the paths in number P, will describe number of the flows over G at a given point in time. A flow x is feasible if it routes 1 unit of flow from s to t. Let The total latency of an edge is denoted $\ell_e(x_e)$ and the total latency of a path is the sum of the latencies of the edges in the path, $\ell_p(X) = \sum_{e\in p} \ell_e(x_e)$ and the average latency of the entire network is $\bar{l}(X) = \sum_{p\in P} x_p \ell_p(X)$ / t.

We use $\ell_p(X)$, $\bar{l}(x)$ and x_p replace those variables in equation(2), and will get:

$$x_p' = \frac{dx_p}{dt} = x_p[l(X) - \bar{l}(X)] \tag{3}$$

Intuitively, Equation (3) shows that paths with below average latency will have more agents switching to them than from them; paths with above average latency will have more agents switching from them than to them. Since the equation (3) contains cubic term x_p^3, according to reference [10], we can deduce that there is no general method for solving this differential equation. But it can be found that under some reasonable conditions stable points of this dynamics will meet with Nash equilibria.

Stability and convergence analysis

In this section, we exploit the stability and convergence characterizations of Nash flow and of P2P ESS in our model.

(a). Eistence of flows at Nash equilibrium

Theorem 1. A P2P instance (G, t, ℓ) evolution game dynamics process $x_p' = x_p[l(X) - \bar{l}(X)]$ with continuous nondecreasing latency function, (a) There is at least one feasible Nash flow in (G, t, ℓ), namely there exits at least one solution $x^*(t)$ in equation (3) and (b) if t x_e , $\tilde{x}_e$ are also Nash flow for (G, t, l), then $\ell_e(x_e) = \ell_e(\tilde{x}_e)$ for every edge e.

Property (b) of theorem 1 means that a flow f is at Nash equilibrium then all s-t edges have equal latency in the P2P routings.

Proof. We assume that x(t) is a solution of equation (3). Let x^* be in equation (3).

$$\begin{aligned}\sum_{p\in P} x'_{p}\big|_{x=x^*} &= \sum_{p\in P} x_p[l(X)-\bar{l}(X)] \\ &= \sum x^* l(X) - \sum x^* \bar{l}(X) \\ &= \sum x^* \cdot \sum x^{*-1} \sum x^* l(X) - \sum x^* \bar{l}(X) \\ &= x^* \cdot \sum \bar{l}(X) - \bar{l}(X) \sum x^* \\ &= \sum x^* \bar{l}(X) - \sum x^* \bar{l}(X) \\ &= 0\end{aligned}$$

$\sum_{p\in P} x'_{p}\big|_{x=x^*} = 0$, therefore $\sum x_p$ is a constant at Nash equilibrium. Since $x_e = \sum_{e\in p} x_p$, then $\ell_e(x_e)$ is also a constant for every edge.

(b). Stability analysis

Theorem 2. The triple (G, t, ℓ) with nonnegative, continuous, and non-decreasing latency function ℓ(*) in evolution game, if flow x is at a Nash equilibrium, then $x\ell(x) < y\ell(x)$ for all y , and hence x is evolutionary stable.

Proof. According to Definition 1, we will give formalized proof about the proof .Let x be a Nash equilibrium. Since x is in a Nash equilibrium, all latencies of used paths belonging to the same commodity are equal and he latency of unused paths is equal or even greater than the latency of used paths. Therefore $x\ell(y) \le y\ell(y)$ for all population y.

(c). Convergence analysis

It has been shown that as time goes to infinity, any initial traffic flows that has support overall paths will eventually converge to a Nash flow. It seems lack of techniques for figuring out how to yield a bound on the time to convergence of Nash flow[8]. So we do not go into specific details of the Nash flow convergence analysis in our model. Since this subsection is focused on convergence properties of our dynamics model, we shall instead give more attention to another result, which bounds the time of convergence to an approximate equilibrium.

To analyze the convergence of the dynamic of P2P imitative model to an approximate equilibrium, we will give some definition and theorem about approximate equilibrium. Definition5 *ε*-approximate equilibrium: Let P_ε be the set of paths that have latency at least $(1+\varepsilon)\bar{l}$, that is $P_\varepsilon = \{p \in P \big| l_p() \ge (1+\varepsilon)l\}$, and let $x_\varepsilon = \sum_{p\in P_\varepsilon} x_p$ be the number of agents using these paths. A population $\vec{x}$ is said to be at a *ε*-approximate equilibrium if and only if $x_\varepsilon \le \varepsilon$.

This definition ensures at such equilibrium that only a small fraction of agents experience latency significantly worse than the minimal latency. In contrast, the definition of a Nash flow requires that all agents experience the same minimal latency.

Theorem 3. The imitative dynamics converge to a ε-approximate equilibrium within time $O(\varepsilon^{-3}\ln(l_{\max}/\hat{l}))$, where $l_{\max}$ mean the maximum latency; $\hat{l}$ means the minimum latency of the traffic flow.

This theorem reveals the upper bound on the time it takes for reach an approximate equilibrium for imitative dynamics model; and also reveals that algorithm based on imitative dynamics model can be achieved within polynomial time. More details about proving the convergence of imitative dynamic to such approximate equilibrium can be referred to works [9,10].

Case study and simulation

The preceding section gives theoretical foundation about our schema. In this section, we carry on a concrete example and algorithm specification to illustrate our aims by simulations.

(a). Evolution Game Algorithm for P2P (EGAP)

Our P2P imitative dynamics model can be deduce to form an optimal algorithm for how to converge to a Nash flow in P2P application and how to reduce bottleneck caused by routing oscillations and selfish routing. Intuitively, we expect each unit of such a flow to travel along the minimum-latency path available to it, where latency is measured with respect to the rest of the flow. Otherwise, this flow would reroute itself on a path with smaller latency.

Recalling that the problem of finding the cost of a flow f is expressed $\ell_p(X) = \sum_{e\in p} \ell_e(x_e)$, note that the problem of finding the minimum latency feasible flow in a P2P network will follow the optimal program:

Min $\sum_{e\in p} C_e(f_e)$

Subject to：

$$\sum f_p = t_i \qquad \forall i \in \{1,\cdots,k\}$$
$$f_e = \sum_{e\in p} f_p \qquad \forall e \in E$$
$$f_p \ge 0 \qquad \forall p \in P$$
$$C_e(f_e) = l_e(f_e) f_e$$

According to those above optimal program description, we will give description of algorithm:

First, when a node wants to start an application session in P2P network, it will determine the route. The node tests the latency of edges firstly. Let the last two minimum latency edges be alternative route edges. Then chose the edge with minimum latency ℓmine as connection path to send connect packet. And then begin P2P application sessions with BitTorrent protocol.

Second, we will use probability sampling model to draw out packets and predict traffic flow amount ratio x_e on the path p. Then algorithm detects the packet drops and revises fitness functions of all strategies by computing δ = $[l_e^{\min} - p^{-1} \cdot \sum_1^p x_e \cdot \hat{l}_e]$. If δ>0, $\ell^{\min}_e$ will be replaced by $\hat{l}_e$ in the s-t path of a peer node; if δ<0, $\ell^{\min}_e$ will still be an edge in the s-t path of a node.

Third, with the evolution game process in the peer routing, all interdomain nodes will gradually achieve a Nash flow statement, where large populations of nodes may converge towards routing equilibrium. If the latency of one node's path is less than the latency of the other agent's path, the agent experiencing higher latency switches to the lower latency path with probability proportional to the difference in latencies. So that flow reroutes itself on a path with smaller latency and the total cost $\sum_{e\in p} C_e(f_e)$ of the paths in P2P network is also in the lowest values.

(b). Simulation and evaluation

Network: Each peer in homogeneous environment has a set of neighbors with which it communicates by message passing. Links are directed. Traffic can however flow in both directions on the links. We consider a P2P network made of total 1024 nodes and each node has 2 Mbps capacity in both directions.

Application: Our implementations for BitTorrent-based application are carried on the java-based peersim simulator, where peers are organized in an overlay network. We follow the simulation methodology and implement in the native

BitTorrent protocol, and we implement additional type of BitTorrent: evolution game algorithm in BitTorrent, in which a peers use imitative mechanism to choose Nash flow paths. Both in native BitTorrent and the improved BitTorrent, trackers choose peers randomly and peers compare their strategies randomly. When a peer completes the download, it reports the event to the tracker. In addition, peers regularly report information such as the total amount of data downloaded so far, the number of bytes that still need to be downloaded, downloaded traffic rate, etc. The tracker keeps all the information in the log files. Hence, we can analyze the tracker log files and retrieve useful information.

Content: In the simulations, 50 unique files of 256 MB are introduced into the system. Each file has been stored at different locations chosen at random. For each experiment, we run the simulation multiple times and the variance in results from multiple rounds is just low, always <10%.

Experiment 1: We take completion time as a metric. It measures the downloading performance of native BitTorrent and Evolution Game Algorithm for P2P (EGAP) and it is also a metric for evaluate the whole system efficiency. It is defined as the total time for a swarm of peers to finish downloading files.

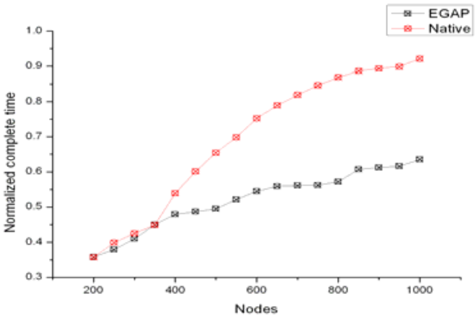

Fig.1 (a). Complete time

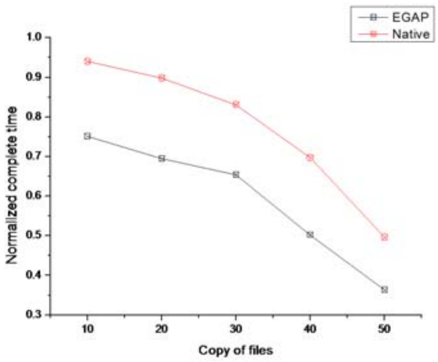

Fig.1 (b). Complete time

In first scenario, we begin the experiment with 200 nodes and 50 files as seeds in the network at first stage then join nodes into the simulation gradually and the number of nodes will be 1024 at final. Fig.1 (a) plots the curve of complete time for two applications. At first time, the difference of complete time between native BitTorrent and evolution game algorithm seems very small. However, with the incensement of the number of nodes, EGAP's result performs much better than native BT. In EGAP, peers reach a approximate Nash equilibrium after a period time, so that they are routing efficiently in the equilibrium state and their complete time do not increase sharply than native BT. Clearly, the reason that EGAP can perform better is due to the evolving process in peers routing, which makes peers select their route by comparing and learning others' strategy in order to find lower latency path and makes peers more interested in achieve Nash flow.

In the second scenario, we begin the experiment with 1024 nodes and 10 files as seeds in the network firstly. Fig.1 (b) plots the results of this scenario. As shown in the figure, EGAP' downloading complete time is always shorter than native one with respective of incensement of the number of file copies. In both two scenarios, the EGAP's complete time is dominant to native one's.

Experiment 2: During every 5-minute time interval, we plots the curve of average latency under different routing paths by computing the traffic equilibria based on the current topology and traffic demands using the approach in EGAP and in native BT in the network with 1024 nodes and 50 copies of file. Given a rate of traffic between each pair of nodes, and a latency function for each edge specifying the time needed to traverse the edge given its congestion. In our experiment we define the ratio of the sum of nodes' FIFO buffering time and the number of nodes as average latency time. Average latency time reflects the downloading performance of applications. As shown in Fig.2, EGAP performs better than native algorithm after a long run time. Considering the average latency time in relation to system expense such as the CPU load and memory cost the nodes have available, it can be seen that the more average latency time nodes have, the more system expense needed to complete downloading tasks. Meanwhile since there is more time needed to empty the queue of packets in native BT and routing requests that make into it will always be added at the end, it takes considerably longer time to process the amount of queries and the time the packet spends waiting to be handled is increased.

That is because the EGAP algorithm would converge to an approximate Nash flow with minimum average latency after nodes evolution game process in routing, while the native algorithm would still in routing chaos situation. In native BT performance such a selfishly motivated routing assignment of traffic to paths will not minimize the average latency; hence, it carries the cost of decreased network downloading performance.

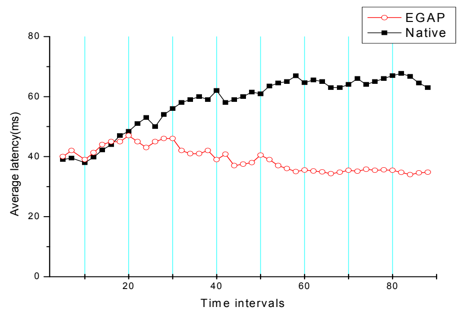

Fig. 2 Latency comparison

Experiment 3: Bottleneck specification. In this scenario, we define bottlenecks typically are at the links of over 10000 packets. Fig.3 and 4 show the bottleneck comparison between native BT and improve BT in the network with 500 nodes and 50 copies of file.

Fig.3 plots the normalized result of bottleneck paths as a function of time during every 5-minute time interval. The normalized result of bottlenecks can be expressed by percentile format of $\sum path_{bottlenecks} / \sum path_{observed}$. $\sum path_{bottlenecks}$ is the numbers of bottleneck path on the snapshot of sampled time and $\sum path_{observed}$ reflects the numbers of paths used on the point of sampled time. This normalized result shows the ratio of bottleneck paths and used paths on the sampled time. As shown in the plot, the native BT application suffered an explicit higher bottleneck than improved one, especially began from the time intervals at 30. Reasons of this phenomenon can be concluded that larger amount of bottleneck traffic caused by disordered search in native BT routing behaviors. Whereas, EGAP can perform a well-ordered routing strategy with small latency first from a global view, so the numbers of bottlenecks can be reduced and the efficiency of the whole network will also be better than the native.

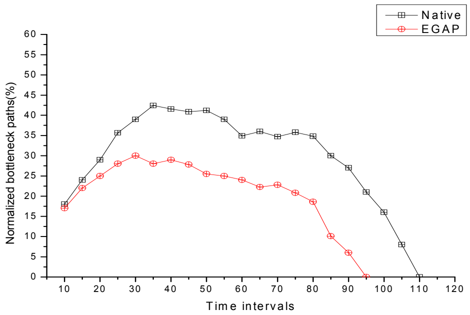

Fig.3 Normalized bottlenecks

Fig.4 plots the total traffic volume of both native BT and EGAP on bottleneck paths. The bottleneck traffic volume means the total traffic volume on bottleneck paths (bottleneck path threshold >10000 packets) of both native BT and EGAP, namely $\sum_{threshold > 10000} traffic_{bottleneck}$.

In Fig.4, as we can see that the total traffic volume of native BT on bottleneck paths is larger than those of EGAP in the whole experiment period. Specifically, native Bit Torrent results in more than 40% higher traffic volume on the bottleneck links at peak minutes. Bottleneck traffic volume curve of native BT increases sharply than that of EGAP at the beginning stage and goes down undulatingly after reaching peak point. That is because selfish and out of order routing in native BT will make all send packets jammed on paths instead of arriving destination unobstructed. Compared against native BitTorrent, EGAP significantly reduces P2P bottleneck traffic, also achieving objective of minimizing whole system's traffic volume.

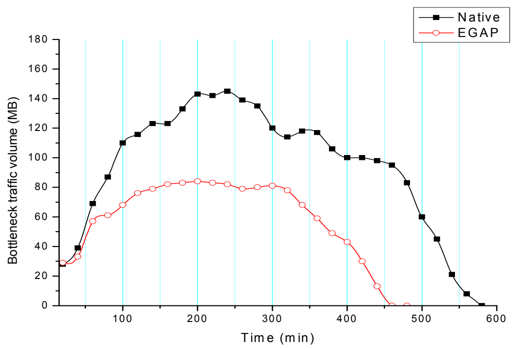

Fig.4 Traffic volume on bottleneck links

Conclusion

In this paper, we presented an evolution game based model for how to eliminate selfish routing behaviors and improve downloading efficiency in P2P networks. Specifically, we proved that a proximate Nash equilibrium exists; under which each peer travels along the minimum-latency path available to it. We also some empirical evaluations for the complete time, the average latency, link occupation and bottleneck traffic etc, which explicitly give us insight on how the network performance is improved by using evolution game algorithm for P2P application.

REFERENCES

[1] Sen S, Wang J. Analyzing Peer-to-Peer Traffic across Large Networks. *Transactions on Networking,*12(2), （2004）219-232.

[2] Karagiannis T, Broido A, Faloutsos M, Claffy KC. Transport layer identification of P2P traffic. *Proc. of the 4th ACM SIGCOMM Conf. on Internet Measurement.* (2004) 121-134

[3] G. Shen, Y. Wang, Y. Xiong, B. Y. Zhao, and Z.-L. Zhang. HPTP: Relieving the tension between ISPs and P2P. *In Proc of IPTPS*（2007）20-29.

[4] O. Saleh, M. Hefeeda. Modeling and caching of peer-to-peer traffic. *Simon Fraser University Technical Report TR* （2006）-11,.

[5] Dabek, F., Cox, R., Kaashoek, F., R. Morris.Vivaldi: A Decentralized Network Coordinate System. *SIGCOMM Comput. Commun. Rev.* 34, 4 (August 2004), 15-26.

[6] Costa, M., Castro, M., Rowstron, A., P. Key. PIC: practical Internet coordinates for distance estimation, *Proceedings. 24th International Conference on Distributed Computing Systems,* (2004)178- 187.

[7] Wang, G., Zhang, B., T. Ng. Towards Network Triangle Inequality Violation Aware Distributed Systems. *In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement,* (2007) 175-188.

[8] Noam Nisan,Tim Roughgarden. Algorithmic Game Theory. *Cambridge University Press*, New York（2007）.

[9] K. Etessami and A. Lochbihler. The computational complexity of evolutionarily stable strategies. *Technical Report TR04-055, Electronic Colloquium on Computational Complexity*(2004).

[10] Jorgen W. Weibull. Evolutionary Game Theory. *MIT press*(1995).

Authors

Fang Zuo received the B.Sc. degree in computer science in 2005 and the M.Phil. degree in applied mathematics in 2008, both from the Henan University. He is now pursuing the Ph.D. degree in East China Normal University, Shanghai, China. His research interests are in P2P networks, distributed multimedia systems, algorithmic game theory and mathematical optimization theory.

Wei Zhang is now a professor of department of Computer Science & Technology in East China Normal University. His research interests are wireless sensor network, computer networks and communication technology, network applications and management, network protocol conformance testing.

Authors: The address of both authors is Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China. E-mail: 52081201007@ecnu.cn (FangZuo); wzhang@cs.ecnu.edu.cn (Wei Zhang).