Impacts of Selfish Behaviors on the Scalability of Hybrid Client–Server and Peer-to-Peer Caching Systems

Youngmi Jin, *Member, IEEE*, George Kesidis, Jinwoo Shin, Fatih Kocak, and Yung Yi, *Member, IEEE*

***Abstract*—This paper considers a hybrid peer-to-peer (p2p) system, a dynamic distributed caching system with an *authoritative* server dispensing contents only if the contents fail to be found by searching an unstructured p2p system. We study the case when some peers may *not* be fully cooperative in the search process and examine the impact of various noncooperative behaviors in the aspect of scalability, more specifically average server load and average peer load as the peer population size increases. We categorize selfish peers into three classes: *impatient peers* that directly query the server without searching the p2p system, *non-forwarders* that refuse to forward query requests, and *non-resolvers* that refuse to share contents. It is shown that in the hybrid p2p system, impatient and/or non-forwarding behaviors prevent the system from scaling well because of the high server load, while the system scales well under the non-resolving selfish peers. Our study implies that the hybrid p2p system does not mandate an incentive mechanism for content sharing, which is in stark contrast to unstructured p2p systems, where incentivizing peers to share contents is known to be a key factor for the system's scalability.**

***Index Terms*—Hybrid, incentive mechanism, peer-to-peer, scalability, selfish behaviors.**

I. Introduction

RECENTLY a hybrid peer-to-peer (p2p) system (also called peer-assisted service) consisting of a few authoritative servers and unstructured peers has received significant attention since it can provide scalable content distribution and overcome weaknesses of purely unstructured p2p systems, including illegal content disseminations [1]–[6]. The measurement study in [1] showed that peer-assisted video-on-demand can significantly reduce server load by shifting the content uploading burden from servers to peers. The streaming capacity of hybrid p2p systems has been analyzed for video-on-demand service and real-time streaming service in [7] and [8]. Ciullo *et al.* analyzed the bandwidth that servers should provide to guarantee ideal service in peer-assisted video-on-demand system [9]. Ioannidis and Marbach in [10] mathematically showed that hybrid p2p systems have good scalability under the assumption that all peers are cooperative. They proved that hybrid p2p systems can alleviate server load significantly and keep load imposed on an individual peer small for a random-walk-based content search, under some conditions on the structures of the peer-connectivity graph family and time-to-live for query resolution.

However, it has been observed that a considerable portion of peers in practice do not cooperate mainly in content sharing and possibly search query forwarding, and such selfish behaviors significantly degrade the performance of content distribution [11], [12]. It is of critical importance to study how the performance of hybrid p2p systems will be when there are selfish peers, which has been underexplored to the best of our knowledge. This paper studies the impact of selfish behaviors on content distribution using local search for a hybrid unstructured p2p and client–server framework consisting of a server and unstructured peers [2]–[5]. We examine how the average load imposed on a server, the average load imposed on an individual peer, and the mean query latency time to get a content change in the presence of selfish peers as the number of peers increases. In this paper, we say that a hybrid p2p system scales well if the average server load and the average peer load do not increase significantly but are kept bounded when the number of peers is increasing.

Note that peers can have various selfish behaviors in terms of what and how. Thus, in this paper we classify selfish behaviors into three categories: 1) *impatience*, meaning direct access to the server without searching other peers; 2) *non-forwarding* that refuses to forward queries; and finally 3) *non-sharing* that refuses to share contents. In our model, the system consists of a server, selfish peers, and altruistic (hence cooperative) peers; some fraction of peers is selfish. Also, depending on how selfish peers behave, we consider two scenarios: a *static* one in which selfish peers always act selfishly when handling queries during the whole stay period in the system, and a *probabilistic* one in which selfish peers opt to act selfishly with some probability

Manuscript received October 08, 2013; revised April 14, 2014 and June 25, 2014; accepted July 16, 2014; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Wang. Date of publication August 26, 2014; date of current version December 15, 2015. This work was supported by the NRF under Grants 2010-0006611 and NRF-2013R1A2A2A01067633 and the NSF CNS under Grants 0916179 and 1152320. A shorter version of this paper appeared in Proc. IEEE INFOCOM 2013. *(Corresponding author: Youngmi Jin.)*

Y. Jin was with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea. She is now with KDDI R&D Labs, Saitama 356-8502, Japan (e-mail:yo-jin@kddilabs.jp).

G. Kesidis is with the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: gik2@psu.edu).

J. Shin and Y. Yi are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: jinwoos@kaist.ac.kr; yiyung@kaist.ac.kr).

F. Kocak was with the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802 USA. He is now with Maxim Integrated, San Jose, CA 95134 USA (e-mail: fwk5027@psu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2014.2347035

when handling queries. The probabilistically selfish peers correspond to those who may want to hide their noncooperativeness to avoid being detected.

This paper investigates how each selfish behavior or their multiple combinations affect the scalability of a hybrid p2p system measured by the traffic load imposed on the server and the load on an individual peer as the number of peers increases.

The main contributions of the paper are as follows.

1) We provide mathematical and numerical analysis of the impacts of various selfish behaviors on the average server load and the average peer load for both static and probabilistic scenarios. In both scenarios, we prove that the scalability of the hybrid p2p system is preserved in the presence of non-resolving (i.e., non-content-sharing) peers under the same conditions for the scalability of fully cooperative hybrid p2p systems. This is in stark contrast to that in purely distributed unstructured p2p systems. However, the scalability does not hold any more in the presence of non-forwarding or impatient peers.
2) For the static scenario, we obtain *closed-form* expressions of the average server loads, the average peer loads, and the average query latency times under various selfish behaviors. This leads to our main results about the scalability of the hybrid p2p system under the static scenario. These studies additionally offer more accurate scalability properties, and they can also be independently useful to other analytical studies of hybrid p2p systems. The main novelty of this analysis lies in a definition of "contact set," which is an extension of the set of peers with the content in the fully cooperative p2p system. This allows us to make mathematical connections between fully cooperative studies in [10] and partially cooperative ones pursued in this paper.
3) The analysis of the probabilistic scenario is much more challenging than that of the static one since the peer selfishness is intermittent and hence the contact set mentioned above becomes highly dynamic. To overcome this issue, instead of obtaining closed forms as in the static scenario, we provide comparison results between the static scenario and the probabilistic one, which suffice to study scalability. Our approach considers "virtual walks" in the query propagation, which provides analytic separation between the query propagation dynamics and the peers' selfish behaviors.

Our results imply that for a hybrid p2p content distribution (or caching) system, an incentive mechanism for content sharing is not mandatory, while an incentive mechanism for impatient peers and/or non-forwarding peers is essential to guarantee scalability. Note that incentive mechanisms for content-sharing have been extensively studied, e.g., [13], and less attention has been paid to other selfish behaviors than non-sharing selfish behaviors [14]. We show that in hybrid p2p systems, non-forwarding and unconditional access to the server cause more dominant increase of the server load than non-content-sharing. Our finding suggests two opposite and arguable points. Since the impact of non-forwarding selfish behaviors is critical, study on incentive mechanisms for *query forwarding* may be important [15]. The other aspect is that query forwarding cost can be regarded negligible since the forwarding cost is much less than content-sharing cost. If forwarding cost is negligible and peers are willing to forward the queries, then in a hybrid p2p system, it suffices to consider an incentive mechanism to prohibit peers from direct accessing the server. Such an incentive mechanism seems much simpler than that for content sharing because an incentive mechanism for content sharing requires a complicated design of fair rewarding and implementation difficulties such as heavy communication overhead load and reliability [13]. Our results show that a hybrid p2p system can be a more practical and efficient content distribution architecture against selfish behaviors than a fully distributed p2p system with an incentive mechanism for content-sharing, when the forwarding cost is negligible.

II. Model

This section describes our model of a hybrid p2p system with a server and many peers. Among many content search mechanisms, we consider the commonly used, lightweight random-walk-based query propagation. Our model is similar to that in [10], except that some peers may be selfish while others are cooperative.

A. Network, Peer Churn, and Query Propagation

Network: We consider a hybrid p2p system that has a single server and N peers. The peers form an unstructured p2p network, and all of them have direct connectivity to the server. The peers constitute an undirected, connected graph $G(V, E)$. The graph G represents a p2p *overlay* network, where one hop in the overlay may correspond to multiple "physical" hops. Denote by d_i the degree of node i, and let $d := \max_{i \in V} d_i$ be the maximum degree. Once the graph G is given, there is an associated random walk, which is a discrete-time Markov chain with transition probability matrix R whose entries are, for all peers $i, j \in V$

$$R_{ij} = \begin{cases} \frac{1}{d_i}, & \text{if } i \neq j,\ (i,j) \in E \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Peer Churn: Peers dynamically enter and leave the system. We assume that as soon as a peer departs the system, a new peer enters the system and replaces the departing peer (this modeling assumption for peer churn is commonly made in literature; see, e.g., [16, Sec. 2.4]). Thus, neither the total number of peers nor its graphical topology G changes. Peers stay in the system for an independent and identically exponentially distributed time with mean $1/\mu$. Peers can generate query requests for the contents in the system. For simplicity, we consider the case where the queries are generated for only one content and those query requests are generated only by newly entering peers with probability p. The query request probability $p = p(N)$ can be a function of the total number of peers N. By Little's formula [17], the mean rate at which peers "arrive" is $N/(1/\mu)$, and the mean load (query rate) at which queries are generated is

$$\text{mean load} = pN\mu. \tag{2}$$

A new peer, who is cooperative, first sends a query to her neighboring peers who further relay the query to other peers. However, there may exist selfish peers who directly access the server. After sending a query to the peers, the (cooperative) peer initiating the query waits for a query resolution response until the given time-to-live $T_{\max}$. If the peer does not get the response by $T_{\max}$, it deems the process of searching the p2p network as failed and sends a new request directly to the server. Note that the server is a last resort to get the content; a peer request launched at the peer group is always resolved, and a peer requesting the content always gets it whether the request is resolved by another peer or by the server. Let $A = A(t) \subset V$ be the set of peers possessing the content at time t. We assume the time–scale separation between churn dynamics and content resolution, i.e., the maximum query response time $T_{\max}$ is negligible compared to the mean peer lifetime $1/\mu$. Then, the churn state $A(t)$ forms a continuous-time Markov process with transition rates depicted in Fig. 1. If a peer, $x \in A$, leaves the system (at rate μ) and the new peer replacing x does not send a request (with probability $1-p$), then the state changes from A to $A - \{x\}$, i.e., at rate $(1-p)\mu$. If $y \in A^c$ leaves the system (at rate μ) and a new peer replacing y sends a request (and acquires the content, with probability p), then the state changes from A to $A \cup \{y\}$ (at rate $p\mu$).

Query Propagation: We assume that queries are propagated by a *continuous-time* random walk that is a lazy version of (1), where the holding time at each peer i is independently exponentially distributed with mean $1/\delta$. A cooperative peer issuing a query sends a query packet to one of its neighbors, which is chosen uniformly at random. A peer that receives the query packet checks whether it has the requested content. If it has the content, it replies and shares the content with the peer who initiated the query. Otherwise, it simply forwards the query packet to one of its neighbors also chosen uniformly at random. However, selfish peers may refuse to forward a query packet or share the content. In the following section, we formally categorize such selfish behaviors. Again, to simplify analysis, we assume that peers can receive the same query packet multiple times (i.e., no "taboo" list is maintained on the packet to avoid cycles or enable reverse-path forwarding of a successfully resolved query). The query propagation dynamics can be thought of as a continuous-time random walk on a *static* p2p network , i.e., no peer leaves or enters the system while a query propagates among them, since it is assumed that the maximum query response time is negligible compared to the peer lifetime. Note that the query propagation is modeled by a transition rate matrix Q of a continuous-time Markov chain

$$Q_{ij} = \frac{\delta}{d_i} \qquad \text{if } i \neq j, (i,j) \in E \tag{3}$$

where a state in the continuous-time Markov process of query propagation is a peer that handles the query packet.

B. Selfish Peers: Behaviors and Scenarios

We assume that a fraction σ of the peers are selfish while the rest are cooperative peers. To formally model selfishness of peers, we classify selfish *behaviors* and consider two *scenarios*, each of which corresponds to being selfish about *what* and *how* peers act selfishly, respectively.

Behaviors: We consider five selfish behaviors as follows.

S1) An *impatient* peer does not send a query request to the p2p system, but instead directly accesses the server to acquire a content without delay.

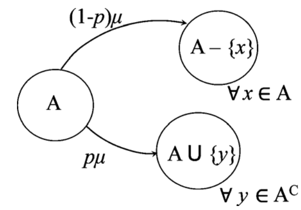


Fig. 1. Continuous-time Markov process $\{A(t)\}$.

S2) A *non-resolving* peer does not share contents, even if it owns them, but merely forwards queries to other peers.

S3) A *non-forwarding* peer does not forward queries and does not possess the queried content.

S4) A *blackhole* peer neither forwards a query nor resolves a query (though the peer may possess the queried content). Note that possessing a content and being a blackhole are "independent" properties of a peer.

S5) A *completely selfish* peer is one that is both impatient and blackhole.

In practice, a peer that does not forward a query is highly likely not to share contents since the cost of content sharing significantly exceeds that of forwarding. However, we separately consider non-forwarding peers who share the contents in order to analyze the pure impact of non-forwarding. When a peer departs the p2p system and a new peer arrives replacing it, the new peer does not necessarily inherit the selfish property of the departing one: It is possible that the departing peer is selfish and the arriving peer is cooperative, or vice versa. We assume that the new entering peer is selfish with probability σ.

Scenarios: Depending on how peers act selfishly, we provide the analysis under two scenarios.

- *Static*: The peer's attribute on selfishness or cooperativeness is sustained until it leaves the system. Hence, *selfish* peers *always* act selfishly whenever they handle a query.
- *Probabilistic*: In this scenario, each *selfish* peer acts selfishly *with probability* $\beta \in (0,1]$ when the peer handles a query request. Thus, the probabilistic behavior for $\beta = 1$ is the same as the static one. This scenario occurs in practice since selfish peers often want to hide their selfish actions by being intermittently selfish.

III. Main Results

This section summarizes and discusses the main results of the impacts of selfish behaviors on the asymptotic average server load and the asymptotic average peer load as N goes to ∞. The complete proofs of theorems stated in this section are provided in Sections IV and V.

A. Prior Work: A Fully Cooperative Case

Before addressing our main results, we briefly state the related results of [10]. Under the assumption that all peers are cooperative, the authors of [10] showed that the average load on the server and the average peer load are bounded as $N \to \infty$ regardless of $p(N)$ (note that p is a function of N) when $T_{\max}(N) = \Theta(N)$[1] and $\{G(N)\}_{N=2}^{\infty}$ is an expander family

[1]A function $f(N) = \Theta(N)$ if there are positive real numbers k_0 and k_1 such that $k_0 N \leq f(N) \leq k_1 N$ for all sufficiently large N.

whose definition is as follows (see [10, Corollaries 1 and 2]). A sequence of graphs, $\{G(N)\}_{N=2}^{\infty}$, is an expander family if

$$\hat{\tau} := \limsup_{N\to\infty} \tau_{G(N)} < \infty$$

(see also [18] and [10, Eq. (3)]) where

$$\tau_G := (1 - \lambda_2^{(R)})^{-1}$$

for a graph $G = (V, E)$ with $|V| = N$, and $\lambda_2^{(R)}$ is the second largest eigenvalue of the transition probability matrix R associated with G [see (1)]. It was evidenced that the overlay graphs of unstructured p2p systems are expanders (e.g., see [19]).

B. Our Results

Once there is a constant fraction of selfish peers, the selfish behaviors generally increase the server load and may compromise the scalability of the hybrid p2p system. Hence, we investigate the scalability of the hybrid p2p system in the aspect of the server load and the peer load.

The increment on the server load depends on selfish behaviors as stated in Theorem 1.

Theorem 1: For both static and probabilistic cases, as $N \to \infty$, the hybrid p2p systems have average server load that is the following:

(a) *bounded* under non-resolving peers, if $\{G(N)\}_{N=2}^{\infty}$ is an expander family and $T_{\max}(N) = \Omega(N)$,[2] regardless of $p(N)$;
(b) *unbounded* under impatient, blackhole, or completely selfish peers when $\lim_{N\to\infty} Np(N) = \infty$;
(c) *unbounded* under non-forwarding peers when $\lim_{N\to\infty} p(N)$ exists and $\lim_{N\to\infty} p(N) < 1$, under the condition that $\lim_{N\to\infty} Np(N) = \infty$.

The condition that $\lim_{N\to\infty} Np(N) = \infty$ in (b) and (c) means that the demand for the content increases with the population size, that is, the content is still popular even if the size of the p2p system gets large. Hence, the results in (b) and (c) imply that the server in the hybrid p2p system cannot support query requests for popular contents as the total number of peers is increasing.

The following theorem states the asymptotic average load per peer in the presence of selfish peers.

Theorem 2: For both static and probabilistic cases, as $N \to \infty$, the hybrid p2p systems have average peer load that is *bounded* under any type of selfish peer, if $\{G(N)\}_{N=2}^{\infty}$ is an expander family and $T_{\max}(N) = O(N)$[3] regardless of $p(N)$.

Combining Theorems 1 and 2, we have the bounded average server load and average peer load by properly designing $T_{\max}(N)$ in the presence of non-resolving selfish peers; the scalability of the hybrid p2p system can be preserved against non-resolving selfish behaviors.

Theorem 3: For both static and probabilistic cases with non-resolving selfish peers, as $N \to \infty$, both the average server load and the average load per peer are bounded for the hybrid p2p systems with non-resolving peers if $\{G(N)\}_{N=2}^{\infty}$ is an expander family and $T_{\max}(N) = \Theta(N)$ regardless of $p(N)$.

Note that under the identical conditions for scalability of a fully cooperative hybrid p2p system, the non-resolving selfish behavior does not destroy the scalability of the p2p system. In the hybrid p2p system, a query request is always resolved either by a peer or by the server, once generated. Hence, a peer generating a query request ultimately owns the content, and there is a sufficient number of (cooperative) peers with the content. However, in a fully distributed p2p system without a server, some query requests may not be resolved because of the absence of a server and selfishness of some peers. The unresolved query requests may increase the number of peers without the content, and as a result, the performance of content distribution can be severely deteriorated by the non-resolving selfish behavior.

C. Implications

We discuss the implications of Theorem 1 with focus on the relation between incentive mechanisms and the server loads. The fact that the server loads are unbounded as the system size increases provides some guidelines on how we should design hybrid p2p systems against selfish peers.

Non-Resolving Peers: As stated in Theorem 1, non-resolving peers do not have critical impact on the server load nor on the peer load for any fixed selfish portion $\sigma < 1$. It implies that a mechanism to incentivize peers to share contents is not mandated in hybrid p2p systems for scalability. This is in contrast to the case of fully distributed p2p systems that typically exert significant efforts to develop nice incentive mechanisms for content sharing [16], [20], [21]. The main reason for this difference is the existence of a dedicated server in hybrid p2p, which ensures to sustain a reasonable degree of content availability, whereas in a fully distributed p2p, the content availability can be worsened over time by non-resolving peers.

Impatient Peers: It is intuitive that the system does not scale well to the increase of the impatient peers because the server load increases in proportion to a σ portion of N peers. However, this unscalability can be easily solved by employing a simple incentive mechanism with the help of the server in hybrid p2p, e.g., enforcing impatient peers to pay a small fee for their impatient direct server access.

Non-Forwarding Peers: In this case, the server load also blows up even for a small portion of non-forwarding peers. Thus, the key of a hybrid p2p system with selfish peers lies in how to incentivize them to forward queries. Two opposite, arguable points can be discussed for non-forwarding peers. The first is that the cost of forwarding (short) query messages may be negligible, compared to that of sharing contents unless resource is scarce (and as a result, users are willing to *forward* query requests). Thus, one may connect Theorem 3 and simple payment-based incentivization for impatient peers to the implication that hybrid p2p systems are generally scalable even in the presence of selfish peers. Note that this contrasts with the case of wireless ad hoc networks with scarce battery and bandwidth resources, where an extensive array of research on forwarding incentivization has been studied, e.g., [22]–[24]. Another point is that despite low query forwarding cost, there may still exist a nonnegligible portion of non-forwarding peers, e.g., simply malicious peers or peers in the competitive p2p systems [25],

[2]A function $f(N) = \Omega(N)$ if there is a real number $k > 0$ such that $f(N) \geq kN$ for all sufficiently large N.

[3]A function $f(N) = O(N)$ if there is a real number $k > 0$ such that $f(N) \leq kN$ for all sufficiently large N.

in which case a scheme for forwarding incentive needs to be applied to the system. However, an incentive mechanism for non-forwarding peers seems to involve some degree of hardness and complexity, especially compared to that for impatient peers. Due to the dedicated server in the hybrid p2p system, developing an incentive mechanism in this case can be easier than that of wireless ad hoc networks, yet it may incur heavy message passing among the server and the peers. Incentive mechanisms for forwarding queries get important as the search for information over social networks becomes popular [15]. To answer more definitely, further experimental studies on existence as well as the portion of non-forwarding peers in hybrid and unstructured p2p systems are necessary.

IV. Static Selfish Behaviors

This section analyzes the effects on the p2p system performance of each type of static selfish behaviors in the aspects of the average server load, the average load per peer, and the average query latency time. When the query request rate is p, these quantities are denoted by the following:

- $\Lambda(p)$: the average load imposed on the server;
- $\rho(p)$: the average load imposed on each peer;
- $\mathsf{D}(p)$: the average query latency time.

The query latency time, $\mathsf{D}(p)$, is the average time taken to get the content after a query is generated, whether a query is resolved either by the server (at $T_{\max}$) or by a peer with the content (within $T_{\max}$).

Separately from $\mathsf{D}(p)$, we define $\mathsf{T}(p)$, which is the average time taken for a query request to terminate, i.e., not to further propagate over the p2p network. This $\mathsf{T}(p)$ will be used to get the average load per peer. To tell the difference between $\mathsf{D}(p)$ and $\mathsf{T}(p)$, we consider when non-forwarding selfish peers are present in the p2p system. There are three cases a query request terminates (where we misuse D and T notation to represent a specific query instance rather than an average) as follows.

i) A query request hits a peer possessing the content at time t_0. In this case, the query is resolved (i.e., terminated) by a peer. Therefore, $\mathsf{D} = \mathsf{T} = t_0$.

ii) A query request hits a non-forwarding peer at time t_0. In this case, the query *stops* at the non-forwarding peer at t_0 (hence $\mathsf{T} = t_0$), stays at the non-forwarding peer until $T_{\max}$, and is then submitted to the server at $T_{\max}$. Hence, ultimately the query is *resolved* by the server at $T_{\max}$, i.e., $\mathsf{D} = T_{\max}$. Therefore, $\mathsf{D} > \mathsf{T}$.

iii) The time-to-live of a query request expires. In this case, the query keeps propagating until $T_{\max}$, and at $T_{\max}$ it is then submitted to the server. Hence, $\mathsf{D} = \mathsf{T} = T_{\max}$.

Before starting our analysis on the p2p system under the selfish behaviors, we first introduce necessary preliminaries for the fully cooperative p2p system in Section IV-A. In the following sections, we will discuss the three quantities including server load for the five selfish behaviors in Section II. From now on, a random walk always means the continuous-time random walk in (3).

A. Fully Cooperative Peers

We state the average server load, the average load per peer, and the query response time in [10] for the case that all of the peers are fully cooperative.

Let $s(p)$ be the probability that a query is resolved by the server. Therefore, for a completely cooperative p2p system (i.e., $\sigma = 0$), the probability that the p2p system resolves a query is $1 - s$, and by (2) the mean load (query rate) on the server is

$$\Lambda(p) = spN\mu.$$

The average load per peer, $\rho(p)$, is defined by the mean number of query messages handled (either forwarded or resolved) by users residing on a vertex of the graph G

$$\rho(p) = \frac{Np\mu H(p)}{N} = p\mu H(p)$$

where $H(p)$ is the average number of hops of the random walk for a single query request until the random walk search finds a peer owning the content within time constraint $T_{\max}$. It can be shown that $\mathsf{T}(p) = \delta H(p)$ (see the proof of [26, Theorem 5.4]). Hence, for the fully cooperative case, the average load per peer is

$$\rho(p) = \frac{p\mu}{\delta}\mathsf{T}(p).$$

Note that for the fully cooperative peers, $\mathsf{T}(p)$ is identical to the average time to find a peer with the content by the random walk search with time constraint; since we do not have case ii) in the above, for the *fully cooperative* case

$$\mathsf{D} \equiv \mathsf{T}.$$

The quantities s and T are obtained by conditioning the set of peers possessing the content, A. The steady-state distribution of the set of peers possessing the content, $A(t)$, is

$$\nu_A(N,p) := \mathsf{P}(A(t) = A) = p^{|A|}(1-p)^{N-|A|} \tag{4}$$

for $A \subseteq V$, which does not depend on μ (peer churn parameter).

For a given set $B \subset V$, $\mathsf{E}_i[T_B]$ denotes the mean hitting time on the set B by the (query propagation) random walk when a query request is generated by peer i. Similarly, $\mathsf{P}_i(T_B > t)$ denotes the probability that a random walk starting from the peer i hits the set B after more than t (seconds).

The expressions for s and T are respectively (see (6)–(9) (proof of Proposition 1) and the proof of Proposition 2 in [10])

$$\begin{aligned} s(p) &= \sum_{A\subseteq V} \mathsf{P}(T_A > T_{\max}|A)\nu_A(N,p) \\ &= \sum_{A\neq V} \nu_A(N-1,p)f_A \qquad (5) \\ \mathsf{T}(p) &= \sum_{A\subseteq V} \mathsf{E}[\mathsf{T}|A]\nu_A(N,p) \\ &= \sum_{A\neq V} \nu_A(N-1,p)g_A \end{aligned}$$

where

$$\begin{aligned} f_A &:= \frac{1}{N}\sum_{j\in A^c} \mathsf{P}_j(T_A > T_{\max}) \\ g_A &:= \frac{1}{N}\sum_{j\in A^c} \mathsf{E}_j\left[\min\{T_A, T_{\max}\}\right]. \qquad (6) \end{aligned}$$

By the definitions, $s(0) = 1$, $\mathsf{T}(0) = T_{\max}$, $s(1) = \mathsf{T}(1) = 0$. We emphasize the following facts.

- g_A and f_A *do not depend on* p.

- $f_A > f_{A\cup\{x\}}$ and $g_A > g_{A\cup\{x\}}$ for $x \notin A$ and $A \subset V$.

Note that $f_\emptyset = 1$ and $g_\emptyset = T_{\max}$, and that g_A and f_A depend on the p2p overlay graph structure and the dynamics of the random walk. In addition, we have found useful properties of $s(p)$ and $\mathsf{T}(p)$ as follows (the proof is in the Appendix).

Lemma 1: $\mathsf{T}(p)$ and $s(p)$ are decreasing convex functions.

Finally, the load per *resolving* peer is

$$\Pi(p) := \frac{(1-s)Np\mu}{pN} = \mu(1-s).$$

B. Impatient Peers (S1)

If a peer is impatient with probability σ (though always cooperatively relaying and resolving) and cooperative with probability $1-\sigma$, then the probability that an "arriving" peer generates a query *to the p2p system* is simply reduced to $(1-\sigma)p$. However, because the impatient peers are assumed to be cooperatively resolving and relaying, impatience does *not* have an effect on the probability of query resolution by the p2p system, $1-s$, and the mean sojourn time of the query in the p2p system, T. Hence, we have

$$s_\sigma^{\mathsf{SIP}}(p) = \sigma + (1-\sigma)s(p)\ (\geq s(p)\ )$$

with $s_0^{\mathsf{SIP}} \equiv s$. The average load per peer is $\rho_\sigma^{\mathsf{SIP}}(p) = \frac{(1-\sigma)p\mu}{\delta}\mathsf{T}(p)$ since the query generation rate to the p2p system is $(1-\sigma)p$. We have $\mathsf{D}_\sigma^{\mathsf{SIP}}(p) = \sigma\cdot 0 + (1-\sigma)\mathsf{T}(p)$ since σ fraction of the amount of queries are directly sent to the server, which do not experience any delay.[4] Summarizing

$$\Lambda_\sigma^{\mathsf{SIP}}(p) = pN\mu s_\sigma^{\mathsf{SIP}}(p) \tag{7}$$
$$\rho_\sigma^{\mathsf{SIP}}(p) = (1-\sigma)\rho(p)$$
$$\mathsf{D}_\sigma^{\mathsf{SIP}}(p) = (1-\sigma)\mathsf{D}(p). \tag{8}$$

Similarly, the load per resolving peer is now $\Pi_\sigma^{\mathsf{SIP}} = (1-\sigma)\Pi(p)$, where $\Pi_0^{\mathsf{SIP}} \equiv \Pi(p) = (1-s)\mu$ is the load per resolving peer for fully cooperative p2p system (with $\sigma = 0$).

C. Non-Resolving Peers (S2)

Now assume σ is the probability that a peer is non-resolving (though patient and relaying), and a peer is cooperative with probability $1-\sigma$. Here, a non-resolving peer (implicitly with content) acts just like a cooperative relaying peer (that does not possess the content).

For this section and Section IV-D, we will use a contact set whose definition is as follows. A *contact set* $\mathcal{C}$ is a set of peers such that if a query request (random walk on the p2p overlay network) reaches any element $j \in \mathcal{C}$, then the query request is terminated.

When there are non-resolving peers, the contact set $\tilde{A}$ is a set of peers that have the content and are cooperative (resolving). The p2p content distribution dynamics can be represented by a Markov process $\tilde{A}(t)$ with transition rates

$$\tilde{A} \to \tilde{A} - \{x\} \qquad \text{with rate } (1-p+p\sigma)\mu\ \forall x \in \tilde{A}$$
$$\tilde{A} \to \tilde{A} \cup \{y\} \qquad \text{with rate } (1-\sigma)p\mu\ \forall y \in \tilde{A}^c.$$

[4]It is assumed that a query request does not experience any delay at the server, when a query request is resolved by the server.

Hence, it follows that

$$s_\sigma^{\mathsf{SNR}}(p) = \sum_{A\neq V} f_A \nu_A\big(N-1, (1-\sigma)p\big)$$
$$\mathsf{T}_\sigma^{\mathsf{SNR}}(p) = \sum_{A\neq V} g_A \nu_A\big(N-1, (1-\sigma)p\big)$$
$$\mathsf{D}_\sigma^{\mathsf{SNR}}(p) = \mathsf{T}_\sigma^{\mathsf{SNR}}(p).$$

Therefore, we have the following proposition.

Proposition 1:

$$\Lambda_\sigma^{\mathsf{SNR}}(p) = pN\mu s_\sigma^{\mathsf{SNR}}(p)$$
$$\rho_\sigma^{\mathsf{SNR}}(p) = \frac{p\mu}{\delta}\mathsf{T}_\sigma^{\mathsf{SNR}}(p)$$
$$\mathsf{D}_\sigma^{\mathsf{SNR}}(p) = \mathsf{T}_\sigma^{\mathsf{SNR}}(p)$$

where

$$s_\sigma^{\mathsf{SNR}}(p) = s((1-\sigma)p)$$
$$\mathsf{T}_\sigma^{\mathsf{SNR}}(p) = \mathsf{T}((1-\sigma)p).$$

From the perspective of query resolution by the p2p system, the mean size of the contact set terminating (and here successfully resolving) the query has changed to $\kappa_\sigma^{\mathsf{SNR}}N$ where

$$\kappa_\sigma^{\mathsf{SNR}} := (1-\sigma)p \tag{9}$$

and, for the fully cooperative case, $\kappa_0^{\mathsf{SNR}} = p$. In the following, κ is the probability that a peer belongs to the contact set $\tilde{A}$ that terminates a query before time-to-live $T_{\max}$, whether or not contacting this set results in the query being successfully resolved by a peer.

Now we can relate the scenario with non-resolving peers to a fully cooperative one as

$$s_\sigma^{\mathsf{SNR}}(p) = s(\kappa_\sigma^{\mathsf{SNR}}) > s(p) \tag{10}$$
$$\mathsf{T}_\sigma^{\mathsf{SNR}}(p) = \mathsf{T}(\kappa_\sigma^{\mathsf{SNR}}) > \mathsf{T}(p)$$

where the inequalities are by Lemma 1. Hence, we have

$$\Lambda_\sigma^{\mathsf{SNF}}(p) \geq \Lambda(p) \quad \rho_\sigma^{\mathsf{SNR}}(p) \geq \rho(p) \quad \mathsf{D}_\sigma^{\mathsf{SNR}}(p) \geq \mathsf{D}(p). \tag{11}$$

The non-resolving selfish behavior has the effect of increasing the average server load, the average peer load, and the average query latency time.

The load per resolving peer is

$$\Pi_\sigma^{\mathsf{SNR}}(p) = \frac{(1-s_\sigma^{\mathsf{SNR}}(p))Np\mu}{(1-\sigma)Np} = \frac{1-s_\sigma^{\mathsf{SNR}}(p)}{(1-\sigma)(1-s)}\Pi(p).$$

Using the convexity of $s(p)$, we can show that $\Pi_\sigma^{\mathsf{SNR}}(p) \geq \Pi(p)$. It is enough to show that $1 - s_\sigma^{\mathsf{SNR}}(p) \geq (1-\sigma)(1-s)$

$$\begin{aligned}&(1-\sigma)(1-s(p))\\ &\quad\leq 1 - s_\sigma^{\mathsf{SNR}}(p)\\ &\quad\Leftrightarrow s((1-\sigma)p) \leq \sigma + (1-\sigma)s(p)\\ &\quad\Leftrightarrow s(\sigma\cdot 0 + (1-\sigma)p) \leq \sigma\cdot s(0) + (1-\sigma)s(p).\end{aligned}$$

Therefore, we know that the load per resolving peer increases because of the non-resolving selfish behavior.

D. Blackhole Peers (S4)

Now suppose that the probability that a peer is a blackhole is σ, and that a peer is cooperative with probability $1-\sigma$. The mean size of the random set of blackholes is $\mathsf{E}\left[A^{\mathsf{SBH}}\right] = \sigma N$. A query will "stop" when it either times out or contacts $\tilde{A} := A^{\mathsf{SBH}} \cup A = A^{\mathsf{SBH}} \cup (A \setminus A^{\mathsf{SBH}})$, where

$$\mathsf{E}[\tilde{A}] = \kappa_\sigma^{\mathsf{SBH}} N \quad \text{and} \quad \kappa_\sigma^{\mathsf{SBH}} := \sigma + (1-\sigma)p.$$

The resulting content-distribution dynamics can be represented by a Markov process $\{\tilde{A}(t)\}$ with transition rates

$$\begin{aligned} &\tilde{A} \to \tilde{A} - \{x\} \quad \text{with rate } (1-p)(1-\sigma)\mu \quad \forall x \in \tilde{A} \\ &\tilde{A} \to \tilde{A} \cup \{y\} \quad \text{with rate } (\sigma + p - \sigma p)\mu \quad \forall y \in \tilde{A}^c. \end{aligned}$$

To get the server load, we need to differentiate whether a query request hits A^{BH} or $A \setminus A^{\mathsf{SBH}}$. Detailed analysis is provided in the proof of Proposition 2. The closed forms of $\Lambda_\sigma^{\mathsf{SBH}}(p)$, $\rho_\sigma^{\mathsf{SBH}}(p)$ and $\mathsf{D}_\sigma^{\mathsf{SBH}}(p)$ are provided below (the proof can be found in the Appendix).

Proposition 2:

$$\begin{aligned} \Lambda_\sigma^{\mathsf{SBH}}(p) &= pN\mu s_\sigma^{\mathsf{SBH}}(p) \\ \rho_\sigma^{\mathsf{SBH}}(p) &= \frac{p\mu}{(1-\sigma)\delta}\mathsf{T}(\kappa_\sigma^{\mathsf{SBH}}) \\ \mathsf{D}_\sigma^{\mathsf{SBH}}(p) &= \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}T_{\max} + \mathsf{T}(\kappa_\sigma^{\mathsf{SBH}})\frac{(1-\sigma)p}{\kappa_\sigma^{\mathsf{SBH}}} \end{aligned}$$

where

$$s_\sigma^{\mathsf{SBH}}(p) = \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}} + s(\kappa_\sigma^{\mathsf{SBH}})\frac{(1-\sigma)p}{\kappa_\sigma^{\mathsf{SBH}}}. \tag{12}$$

Since blackhole peers have two selfish behaviors, non-resolving and non-forwarding, we intuitively expect the server load with blackhole peers is bigger than that with non-resolving peers. The following corollary shows that this intuition is true (the proof is in the Appendix).

Corollary 1:

$$\begin{aligned} \Lambda_\sigma^{\mathsf{SBH}}(p) &\geq \Lambda_\sigma^{\mathsf{SNR}}(p) \geq \Lambda(p) \\ \rho_\sigma^{\mathsf{SBH}}(p) &\leq \rho(p) \leq \rho_\sigma^{\mathsf{SNR}}(p) \\ \mathsf{D}_\sigma^{\mathsf{SBH}}(p) &\geq \mathsf{D}_\sigma^{\mathsf{SNR}}(p) \geq \mathsf{D}(p). \end{aligned}$$

From Corollary 1, the presence of blackhole selfish peers increases the average server load and the query latency time, but decreases the average load per peer.

The mean query rate on the (non-blackhole) resolving peers satisfies

$$\begin{aligned} \Pi_\sigma^{\mathsf{SBH}}(p) &= \frac{\left(1 - s_\sigma^{\mathsf{SBH}}(p)\right)pN\mu}{(1-\sigma)pN} \\ &= \frac{1 - s_\sigma^{\mathsf{SBH}}(p)}{(1-\sigma)(1-s(p))}\Pi(p) \left(\leq \Pi_\sigma^{\mathsf{SNR}}(p) \right). \end{aligned} \tag{13}$$

Intuitively, this comes from the observation that more query requests are terminated without being resolved by peers (hence fewer requests arrive at peers possessing the content) in the presence of blackhole selfish peers than in the presence of non-resolving peers.

E. Non-Forwarding Peers (S3)

The effect of non-forwarding behavior is similar to that of blackhole behavior. Now suppose that the probability that a peer is non-forwarding is σ, and that a peer is cooperative with probability $1-\sigma$. The contact set that terminates a query is the set of peers that are selfish or that have the content. The mean size of the contact set is $\kappa_\sigma^{\mathsf{SNF}} N$, where

$$\kappa_\sigma^{\mathsf{SNF}} := \sigma(1-p) + p = \kappa_\sigma^{\mathsf{SBH}}. \tag{14}$$

In the presence of non-forwarding peers, we have the following proposition (see the Appendix for the proof).

Proposition 3:

$$\begin{aligned} \Lambda_\sigma^{\mathsf{SNF}}(p) &= pN\mu s_\sigma^{\mathsf{SNF}}(p) \\ \rho_\sigma^{\mathsf{SNF}}(p) &= \frac{p\mu}{\delta} \cdot \frac{\mathsf{T}(\kappa_\sigma^{\mathsf{SNF}})}{(1-\sigma)(1-p)+p} \\ \mathsf{D}_\sigma^{\mathsf{SNF}}(p) &= \frac{\sigma(1-p)}{\kappa_\sigma^{\mathsf{SNF}}}T_{\max} + \mathsf{T}(\kappa_\sigma^{\mathsf{SNF}})\frac{p}{\kappa_\sigma^{\mathsf{SNF}}} \end{aligned}$$

where

$$s_\sigma^{\mathsf{SNF}}(p) = \frac{\sigma(1-p)}{\kappa_\sigma^{\mathsf{SNF}}} + s(\kappa_\sigma^{\mathsf{SNF}})\frac{p}{\kappa_\sigma^{\mathsf{SNF}}}.$$

Furthermore, we observe that

$$\begin{aligned} s_\sigma^{\mathsf{SBH}}(p) \geq s_\sigma^{\mathsf{SNF}}(p) &= \frac{\sigma(1-p)}{\kappa_\sigma^{\mathsf{SNF}}}s(0) + s(\kappa_\sigma^{\mathsf{SNF}})\frac{p}{\kappa_\sigma^{\mathsf{SNF}}} \\ &\geq s\left(\frac{\sigma(1-p)}{\kappa_\sigma^{\mathsf{SNF}}} \cdot 0 + \frac{p}{\kappa_\sigma^{\mathsf{SNF}}} \cdot \kappa_\sigma^{\mathsf{SNF}}\right) = s(p) \end{aligned}$$

where we use $s(0) = 1$ for the first equality and the second inequality holds because of the convexity of $s(p)$. Similarly

$$\begin{aligned} \mathsf{D}_\sigma^{\mathsf{SBH}}(p) \geq \mathsf{D}_\sigma^{\mathsf{SNF}}(p) &= \mathsf{T}(0)\frac{\sigma(1-p)}{\kappa_\sigma^{\mathsf{SNF}}} + \mathsf{T}(\kappa_\sigma^{\mathsf{SNF}})\frac{p}{\kappa_\sigma^{\mathsf{SNF}}} \\ &\geq \mathsf{T}\left(\frac{\sigma(1-p)}{\kappa_\sigma^{\mathsf{SNF}}} \cdot 0 + \frac{p}{\kappa_\sigma^{\mathsf{SNF}}} \cdot \kappa_\sigma^{\mathsf{SNF}}\right) = \mathsf{T}(p) \\ &= \mathsf{D}(p) \end{aligned}$$

since $\mathsf{T}(0) = T_{\max}$, $\mathsf{T}(p) = \mathsf{D}(p)$ (for $\sigma = 0$) and the convexity of $\mathsf{T}(p)$. For the average load per peer, $\rho_\sigma^{\mathsf{SNF}}(p) \leq \rho_\sigma^{\mathsf{SBH}}(p)$ holds since $(1-\sigma)(1-p)+p \geq 1-\sigma$. This can be intuitively explained by the observation that there are more (cooperative) resolving peers in the presence of non-forwarding selfish peers than in the presence of blackhole selfish ones. Summarizing all these, we have the following corollary.

Corollary 2:

$$\begin{aligned} \Lambda_\sigma^{\mathsf{SBH}}(p) &\geq \Lambda_\sigma^{\mathsf{SNF}}(p) \geq \Lambda(p) \\ \rho_\sigma^{\mathsf{SNF}}(p) &\leq \rho_\sigma^{\mathsf{SBH}}(p) \leq \rho(p) \leq \rho_\sigma^{\mathsf{SNR}}(p) \\ \mathsf{D}_\sigma^{\mathsf{SBH}}(p) &\geq \mathsf{D}_\sigma^{\mathsf{SNF}}(p) \geq \mathsf{D}(p). \end{aligned}$$

The analogous version of (13) holds as well

$$\Pi_\sigma^{\mathsf{SNF}}(p) = \frac{1 - s_\sigma^{\mathsf{SNF}}(p)}{(1-\sigma)(1-s(p))}\Pi(p) \left(\leq \Pi_\sigma^{\mathsf{SNR}}(p) \right).$$

Hence, obviously

$$\Pi_\sigma^{\mathsf{SBH}}(p) \leq \Pi_\sigma^{\mathsf{SNF}}(p) \leq \Pi_\sigma^{\mathsf{SNR}}(p).$$

F. Completely Selfish Peers (S5)

A completely selfish peer is both impatient and a blackhole. Considering the separate cases above, the probability that a query request is resolved by the server is

$$\begin{aligned} s_\sigma^{\mathsf{SCS}}(p) &= \sigma + (1-\sigma)s_\sigma^{\mathsf{SBH}}(p) \\ &= s_\sigma^{\mathsf{SBH}}(p) + \sigma(1 - s_\sigma^{\mathsf{SBH}}(p)). \end{aligned}$$

For the average peer load and the average query latency time, note that the case of completely selfish peers is completely identical to the case of blackhole peers except that only $1-\sigma$ fraction of traffic load is imposed on the p2p system. Summarizing, we have the following proposition.

Proposition 4:

$$\begin{aligned} &\Lambda_\sigma^{\mathsf{SCS}}(p) = Np\mu s_\sigma^{\mathsf{SCS}}(p) \\ &\rho_\sigma^{\mathsf{SCS}}(p) = (1-\sigma)\rho_\sigma^{\mathsf{SBH}}(p) = \frac{p\mu}{\delta}\mathsf{T}(\kappa_\sigma^{\mathsf{SBH}}) \\ &\mathsf{D}_\sigma^{\mathsf{SCS}}(p) = (1-\sigma)D_\sigma^{\mathsf{SBH}}(p). \end{aligned}$$

Simple algebra results in the following corollary.

Corollary 3:

$$\begin{aligned} &\Lambda_\sigma^{\mathsf{SCS}}(p) \geq \Lambda_\sigma^{\mathsf{SBH}}(p) \geq \Lambda_\sigma^{\mathsf{SNF}}(p) \text{ or } \Lambda_\sigma^{\mathsf{SNR}}(p) \geq \Lambda(p) \\ &\rho_\sigma^{\mathsf{SCS}}(p) \leq \rho_\sigma^{\mathsf{SNF}}(p) \leq \rho_\sigma^{\mathsf{SBH}}(p) \leq \rho(p) \leq \rho_\sigma^{\mathsf{SNR}}(p) \\ &\mathsf{D}_\sigma^{\mathsf{SBH}}(p) \geq \mathsf{D}_\sigma^{\mathsf{SCS}}(p) \geq \mathsf{D}_\sigma^{\mathsf{SNF}}(p) \geq \mathsf{D}(p). \end{aligned}$$

G. Proof of Theorem 1 for Static Selfish Behaviors

This section provides the proof of Theorem 1 under static selfish behaviors.

S1) Under impatient peers, by (7) the average load on the server is greater than $\sigma p N\mu$, which diverges as $Np \to \infty$.

S2) Under non-resolving peers, the server load is

$$\begin{aligned} pN\mu s(\kappa_\sigma^{\mathsf{SNR}}) &= pN\mu s\Big((1-\sigma)p\Big) \\ &= \frac{1}{(1-\sigma)}\{p'N\mu s(p')\} \end{aligned}$$

where $p' = (1-\sigma)p$ and the first equality is due to Proposition 1. By [10, Corollary 1], $p'N\mu s(p')$ is bounded if $\{G(N)\}_{N=2}^{\infty}$ is an expander graph family and $T_{\max}(N) = \Omega(N)$ as $N \to \infty$. Hence, the server load $s(\kappa_\sigma^{\mathsf{SNR}})pN\mu$ is bounded regardless of p with the assumptions.

S3) To derive a contradiction, suppose that the server load is bounded under non-forwarding peers. Namely, for some $K < \infty$

$$\lim_{N\to\infty} \mu Nps_\sigma^{\mathsf{SNF}}(p) < K.$$

Then, we observe that

$$\begin{aligned} \mu Np &\leq \frac{K}{s_\sigma^{\mathsf{SNF}}(p)} \leq K\frac{\sigma(1-p)+p}{\sigma(1-p)} \\ &= K\left(1 - \frac{1}{\sigma} + \frac{1}{\sigma(1-p)}\right) \end{aligned} \tag{15}$$

where the second inequality holds by Proposition 3. Thus, when $\lim_{N\to\infty} p(N) < 1$, the right-hand side of (13) has a finite limit, which contradicts that μNp is unbounded. Therefore, $\lim_{N\to\infty} pN\mu s_\sigma^{\mathsf{SNF}}(p) = \infty$.

S4) Under blackhole peers, the server load diverges as $Np \to \infty$ because

$$\mu Nps_\sigma^{\mathsf{SBH}}(p) \geq \mu Np\frac{\sigma}{\sigma + (1-\sigma)p} > \mu Np\sigma$$

where the first inequality is due to Proposition 2.

S5) Under completely selfish peers, the server load is unbounded simply because the completely selfish case is for the corresponding system with blackhole peers.

H. Proof of Theorem 2 for Static Selfish Behaviors

This section provides the proof of Theorem 2 under static selfish behaviors.

S1) Under impatient peers, by (8) the average load on the peer is smaller than $\rho(p)$; $\rho_\sigma^{\mathsf{SIP}}(p) = (1-\sigma)\rho(p)$, which is bounded under the conditions of Theorem 2 since $\rho(p)$ is bounded with the same conditions.

S2) Under non-resolving peers, the average load per peer is

$$\begin{aligned} \rho_\sigma^{\mathsf{SNR}}(p) &= \frac{p\mu}{\delta}\mathsf{T}\Big((1-\sigma)p\Big) \\ &= \frac{1}{(1-\sigma)}\Big(\frac{p'\mu}{\delta}\mathsf{T}(p')\Big) \end{aligned}$$

where $p' = (1-\sigma)p$ and the first equality is due to Proposition 1. By [10, Corollary 2], $\frac{p'\mu}{\delta}\mathsf{T}(p') = \mathsf{T}((1-\sigma)p)$ is bounded if $\{G(N)\}_{N=2}^{\infty}$ is an expander graph family and $T_{\max}(N) = O(N)$ as $N \to \infty$. Hence, the average peer load $\rho_\sigma^{\mathsf{SNR}}(p)$ is bounded regardless of p with the assumptions.

S3, S4, and S5) This immediately holds by Corollary 3 and the boundedness of $\rho_\sigma^{\mathsf{SNR}}(p)$ shown above under the assumptions.

V. Probabilistically Selfish Behaviors

This section considers probabilistically selfish behaviors. We recall the definition: An arriving peer is selfish with probability σ and will thereafter behave selfishly when handling queries only with probability β. In other words, a selfish peer may behave differently when handling the same query more than once. Recall that our random walk is assumed memoryless. Note that this is another way to model how a node may (selfishly or maliciously) attempt deplete a query's time-to-live $T_{\max}$. A motivation for acting selfish only in such a probabilistic manner is to avoid detection and classification as a noncooperative peer.

We analyze the asymptotic server load and the asymptotic load per peer for various probabilistically selfish peers that act selfishly [according to types of (*S2*)–(*S4*)] as in Section IV. A similar strategy to what we used to establish the asymptotic server load for static selfish peers does not work for probabilistically selfish peers. Instead, we establish the comparison results for the asymptotic server load and the asymptotic peer load (proofs of Propositions 5 and 6 are in the Appendix).

Proposition 5: If $\sigma_1\beta_1 = \sigma_2\beta_2$ and $\beta_1 \leq \beta_2$, then

$$\begin{aligned} &\text{(a)} \quad \Lambda_{\sigma_1,\beta_1}^{\mathsf{PNR}}(p) \leq \Lambda_{\sigma_2,\beta_2}^{\mathsf{PNR}}(p)\ (\leq \Lambda_{\sigma_1\beta_1}^{\mathsf{SNR}}(p)) \\ &\text{(b)} \quad \Lambda_{\sigma_1,\beta_1}^{\mathsf{PNF}}(p) \geq \Lambda_{\sigma_2,\beta_2}^{\mathsf{PNF}}(p)\ (\geq \Lambda_{\sigma_1\beta_1}^{\mathsf{SNF}}(p)) \\ &\text{(c)} \quad \Lambda_{\sigma_1,\beta_1}^{\mathsf{PBH}}(p) \geq \Lambda_{\sigma_2,\beta_2}^{\mathsf{PBH}}(p)\ (\geq \Lambda_{\sigma_1\beta_1}^{\mathsf{SBH}}(p)). \end{aligned}$$

Since we have

$$\Lambda_{\sigma}^{\mathsf{SNR}}(p)=\Lambda_{\sigma,1}^{\mathsf{PNR}}(p) \quad \Lambda_{\sigma}^{\mathsf{SNF}}(p)=\Lambda_{\sigma,1}^{\mathsf{PNF}}(p) \quad \Lambda_{\sigma}^{\mathsf{SBH}}(p)=\Lambda_{\sigma,1}^{\mathsf{PBH}}(p)$$

the conclusions of Theorem 1 for probabilistically selfish behaviors can be derived using Proposition 5 and Theorem 1 for static selfish behaviors.

Proposition 6: If $\sigma_1\beta_1 = \sigma_2\beta_2$ and $\beta_1 \leq \beta_2$, then

$$\begin{aligned} &\text{(a)} \qquad \rho_{\sigma_1,\beta_1}^{\mathsf{PNR}}(p) \leq \rho_{\sigma_2,\beta_2}^{\mathsf{PNR}}(p) \ (\leq \rho_{\sigma_1\beta_1}^{\mathsf{SNR}}(p)) \\ &\text{(b)} \qquad \rho_{\sigma_1,\beta_1}^{\mathsf{PNF}}(p) \leq \rho_{\sigma_2,\beta_2}^{\mathsf{PNF}}(p) \ (\leq \rho_{\sigma_1\beta_1}^{\mathsf{SNF}}(p)) \\ &\text{(c)} \qquad \rho_{\sigma_1,\beta_1}^{\mathsf{PBH}}(p) \leq \rho_{\sigma_2,\beta_2}^{\mathsf{PBH}}(p) \ (\leq \rho_{\sigma_1\beta_1}^{\mathsf{SBH}}(p)). \end{aligned}$$

Since we have

$$\rho_{\sigma}^{\mathsf{SNR}}(p) = \rho_{\sigma,1}^{\mathsf{PNR}}(p) \quad \rho_{\sigma}^{\mathsf{SNF}}(p) = \rho_{\sigma,1}^{\mathsf{PNF}}(p) \quad \rho_{\sigma}^{\mathsf{SBH}}(p) = \rho_{\sigma,1}^{\mathsf{PBH}}(p)$$

the conclusions of Theorem 2 for probabilistically selfish peers hold by Proposition 6 and the results of static selfish behaviors in Theorem 2.

VI. Numerical Study

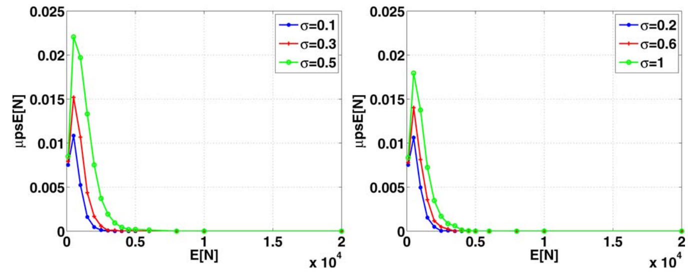
This section shows our numerical experiments in which we relaxed several modeling assumptions in Section II for the validity of our analytical results. The modeling assumptions and their relaxations are as follows.

- The overlay graph structure and total population size are fixed by assuming that whenever a node departs the system, a newly entering peer replaces it: In our simulations, there is no such restriction on the departure and entering of nodes. Hence, the overlay graph structure and the number of peers vary as time evolves in the simulations.
- Queries are generated only when new peers enter the system: In our simulations, queries are generated during the lifetime of a peer, not just upon arrival.

To create the overlay graphs (of peers) of an expander family, we used an algorithm proposed in [27] that is known to generate expander graphs with high probability (we refer the reader to [27] and [10, Section 5.1]). In our simulations, the overlay graphs consist of two Hamilton cycles [28], and each user in the graph has degree 4 in average.

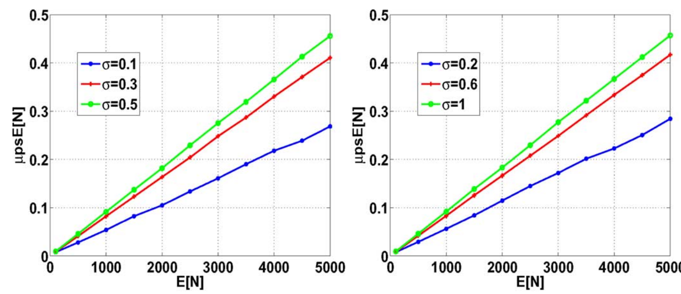
For each plot, $p(N) = 0.1$ and $T_{\max}(N) = N/20$. Constant $p(N)$ $(= 0.1)$ implies that query requests are frequently generated in the sense that the content is popular even when the population size is increasing. Hence, we investigate the impact of selfish behaviors when the demand of the content is high. Users arrive according to a Poisson process with rate λ and depart with rate $\mu = 0.001$. To scale the system, we use different arrival rates λ and do simulations with each value of λ. Since the total number of peers varies with time, the system size is represented by the average number of peers, which is $\mathsf{E}[N] = \lambda/\mu$ by Little's Law. Due to space limitation, we provide the experimental results of only non-resolving and blackhole peers.

Fig. 2 depicts the bounded asymptotic server load of the p2p system with non-resolving peers. It shows that as in Theorem 1, the server load decreases with $\mathsf{E}[N] > 1000$ and is bounded ultimately. We took $\beta = 0.5$ for the case of probabilistic (blackhole



(a) (b)

Fig. 2. Server load with non-resolving peers. (a) SNR. (b) PNR with $\beta = 0.5$.



(a) (b)

Fig. 3. Server load with blackhole peers. (a) SBH. (b) PBH with $\beta = 0.5$.

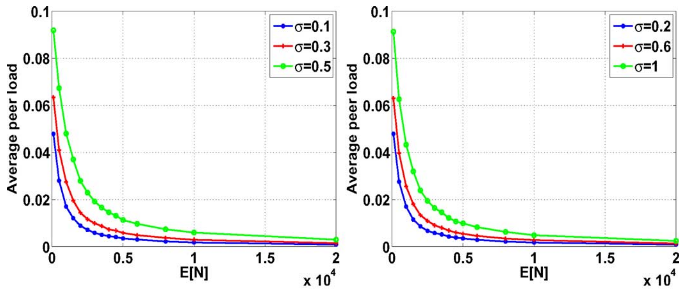
or non-resolving) peers for both cases. Fig. 2 shows that $s_{\sigma,\beta}^{\mathsf{PNR}} \leq s_{\phi}^{\mathsf{SNR}}$ for $\phi = \sigma \times \beta$, which is consistent with Proposition 5.

The above result, the bounded average server load, still holds when $p(N)$ is decreasing with N, i.e., when queries are not frequently generated as long as $T_{\max}(N)$ increases faster than or proportional to N. However, the average server load is not bounded any more if $T_{\max}(N)$ changes slower than N, for example, $T_{\max}(N) = \log N$.

In Fig. 3, the total loads on the server, $\mu pNs_{\sigma}^{\mathsf{SBH}}$ and $\mu pNs_{\sigma,\beta}^{\mathsf{PBH}}$, are depicted for static and probabilistic blackhole peers, respectively. Here, we observe that the load on the server is unbounded in both plots as Theorem 1 suggests. We took $\beta = 0.5$ for the case of probabilistic (blackhole or non-resolving) peers. Fig. 3 shows that $s_{\sigma,\beta}^{\mathsf{PBH}} \geq s_{\phi}^{\mathsf{SBH}}$ for $\phi = \sigma \times \beta$ in Proposition 5.

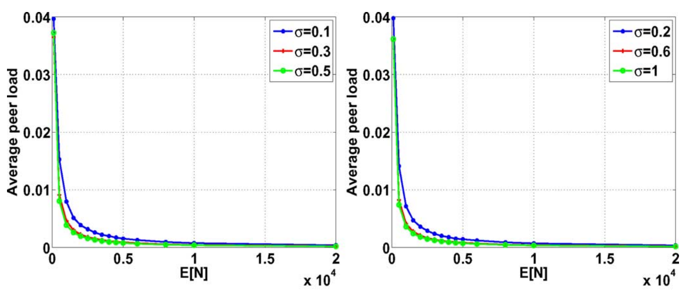
Note that these results are for frequently generated queries, i.e., constant $p(N)$. When queries are not frequently generated, the average server load may be bounded even for the existence of blackhole selfish peers. Indeed, for $p(N) = 1/N$, the average server load is *bounded* with *static blackhole* peers existence, which can be easily shown by using Proposition 2.

Figs. 4 and 5 show how the asymptotic average load per peer changes for both non-resolving and blackhole peers, respectively, as the number of peers is increasing. As in Theorem 2, both plots exhibit bounded average peer load. Note that in Fig. 4, the average load per peer for non-resolving peers is increasing, while in Fig. 5, the average load per peer is decreasing as σ is increasing. The rationale behind this is as follows. When blackhole selfish peers exist, queries hitting blackhole peers do not propagate further. This implies that less forwarding takes places with big σ than with small σ. Therefore, the average peer load is decreasing as σ is increasing. Meanwhile in the case



(a) (b)

Fig. 4. Peer load with non-resolving peers: nonregular graphs. (a) SNR. (b) PNR with $\beta = 0.5$.



(a) (b)

Fig. 5. Peer load with blackhole peers: nonregular graphs. (a) SBH. (b) PBH with $\beta = 0.5$.

of non-resolving peers, a query propagates until it hits a *cooperative* peer possessing the content. Hence, in the presence of non-resolving selfish peers, big σ implies longer search time or more forwarding for a cooperative peer owning the content than with small σ; the average load per peer is increasing with σ in the presence of non-resolving selfish peers.

In the real implementation of a scalable p2p system, $T_{\max}(N)$ is the design parameter. We need to set $T_{\max}(N) = \Omega(N)$ for the bounded average server load and to set $T_{\max}(N) = O(N)$ for the bounded peer load as in Theorems 1 and 2. Hence, combined both, $T_{\max}(N)$ is to be linear with N for scalable hybrid p2p in the real implementation. Our results shown in Fig. 2 are the case when there is only one single server. If there are multiple servers, then the average server load would be reduced more than that of a single-server case. In the real implementation, multiple servers are recommended rather than a single server for the failure of a server and the low average server load and low latency.

VII. Conclusion

This paper has analyzed the impact of selfish behaviors on the performance of content distribution of an unstructured hybrid p2p system, which exhibits good scalability for an expander graph family when all N peers are cooperative and the time-to-live of a query request is designed as $\Theta(N)$. We classified different selfish behaviors and analyzed mathematically and numerically how the asymptotic average server load and the asymptotic average peer load change by the selfish behaviors. Our analysis revealed that non-resolving selfish behavior does not compromise the scalability, while selfish behaviors of non-forwarding and direct-accessing the server cause the server load of the hybrid p2p system to increase with the number of peers N. These results suggest that a hybrid system can be designed to be scalable without an incentive mechanism for content sharing. However, the system does need incentive mechanisms for query request forwarding and access to peers for contents.

Appendix
Proofs

Proof of Lemma 1: By direct differentiation

$$\begin{aligned} s'(p) = & \sum_{B:1\le|B|\le N-1} f_B|B|p^{|B|-1}(1-p)^{N-1-|B|} \\ & \sum_{A:0\le|A|\le N-2} (N-1-|A|)f_A p^{|A|}(1-p)^{N-2-|A|} \\ = & \sum_{k=0}^{N-2} p^k(1-p)^{N-2-k}H(k) \end{aligned}$$

where

$$H(k) = \sum_{B:|B|=k+1}(k+1)f_B - \sum_{A:|A|=k}(N-1-k)f_A.$$

To show that $H(k) < 0$ for $0 \le k \le N-2$, first note that for any $j \in A^c$, $x \in A^c$, $j \ne x$

$$\mathsf{P}_j(T_A > T_{\max}) \ge \mathsf{P}_j(T_{A\cup\{x\}} > T_{\max}).$$

Then, for any $x \in A^c$

$$\begin{aligned} Nf_A &= \mathsf{P}_x(T_A > T_{\max}) + \sum_{j\in A^c-\{x\}} \mathsf{P}_j(T_A > T_{\max}) \\ &\ge \mathsf{P}_x(T_A > T_{\max}) + \sum_{j\in A^c-\{x\}} \mathsf{P}_j(T_{A\cup\{x\}} > T_{\max}) \\ &= \mathsf{P}_x(T_A > T_{\max}) + Nf_{A\cup\{x\}}. \end{aligned}$$

Summing both sides over all $x \in A^c$, we have

$$\begin{aligned} \sum_{x\in A^c} Nf_A &\ge Nf_A + \sum_{x\in A^c} Nf_{A\cup\{x\}} \\ &\Rightarrow (N-|A|-1)f_A \ge \sum_{x\in A^c} f_{A\cup\{x\}} \\ &\Rightarrow \sum_{A:|A|=k}(N-k-1)f_A \ge \sum_{B:|B|=k+1}(k+1)f_B. \end{aligned}$$

Similarly, by substituting f_A by g_A and $\mathsf{P}_j(T_A > T_{\max})$ by $\mathsf{E}_j \min\{T_A, T_{\max}\}$, we can show that $\mathsf{T}'(p) < 0$.

The proof of the convexity is in Corollary 4 right after the proof of Corollary 1. ∎

Proof of Proposition 2: By conditioning on $\tilde{A} = A^{\mathsf{SBH}} \cup A$, we get

$$1 - s_\sigma^{\mathsf{SBH}}(p) = \sum_{\tilde{A}\subseteq V} h_{\tilde{A}}(\sigma,p)\nu_{\tilde{A}}(N,\kappa_\sigma^{\mathsf{SBH}}) \tag{16}$$

where $h_{\tilde{A}}(\sigma,p) := \mathsf{P}(T_{A\setminus A^{\mathsf{SBH}}} < \min\{T_{\max}, T_{A^{\mathsf{SBH}}}\})$ (and we suppressed indication of conditioning on the initial (querying) peer on $\tilde{A}^c$). Let $X(t)$ be the peer handling the query at time t, so that

$$\begin{aligned} h_{\tilde{A}}(\sigma,p) &= \mathsf{P}(T_{\tilde{A}} < T_{\max},\ X(T_{\tilde{A}}) \in A \setminus A^{\mathsf{SBH}}) \\ &= \frac{(1-\sigma)p}{\kappa_\sigma^{\mathsf{SBH}}}\,\mathsf{P}(T_{\tilde{A}} < T_{\max}) \end{aligned} \tag{17}$$

since the event of $T_{\tilde{A}} < T_{\max}$ and the event of $X(T_{\tilde{A}}) \in A^{\mathsf{SBH}}$ are independent. Therefore

$$1 - s_\sigma^{\mathsf{SBH}}(p)r = \frac{(1-\sigma)p}{\kappa_\sigma^{\mathsf{SBH}}} \sum_{\tilde{A}\subseteq V} \mathsf{P}(T_{\tilde{A}} < T_{\max})\nu_{\tilde{A}}(N, \kappa_\sigma^{\mathsf{SBH}}) = \frac{(1-\sigma)p}{\kappa_\sigma^{\mathsf{SBH}}}\Big(1 - s(\kappa_\sigma^{\mathsf{SBH}})\Big). \quad (18)$$

Then, we have obtained $s_\sigma^{\mathsf{SBH}}(p)$ and $\Lambda_\sigma^{\mathsf{SBH}}(p)$.

Now we consider the average load per peer. Note that blackhole peers do not handle query requests; they do not forward a query nor share content. Hence, the average number of peers who handle a query is $(1-\sigma)N$. Therefore, the load imposed on a peer is

$$\rho_\sigma^{\mathsf{SBH}}(p) = \frac{Np\mu}{(1-\sigma)N} H_\sigma^{\mathsf{SBH}}(p)$$

where $H_\sigma^{\mathsf{SBH}}(p)$ is the average number of hops of the random walk search for a single query request when there are blackhole selfish peers. As in the fully cooperative case

$$\delta \cdot H_\sigma^{\mathsf{SBH}}(p) = \mathsf{T}_\sigma^{\mathsf{SBH}}(p) = \mathsf{T}(\kappa_\sigma^{\mathsf{SBH}})$$

where $\mathsf{T}_\sigma^{\mathsf{SBH}}(p)$ is the average time of a query request to terminate whether it is resolved or stopped by hitting a blackhole peer and $\mathsf{T}_\sigma^{\mathsf{SBH}}(p) = (\mathsf{T}(\kappa^{\mathsf{SBH}})$ because of the p2p system's content-distribution dynamics and $\mathsf{E}[\tilde{A}] = \kappa_\sigma^{\mathsf{SBH}} N$.

Finally, by conditioning on $\tilde{A}$, we have

$$\mathsf{D}_\sigma^{\mathsf{SBH}}(p) = \sum_{\tilde{A}\subseteq V} \mathsf{E}[\mathsf{D}_\sigma^{\mathsf{SBH}}|\tilde{A}]\nu_{\tilde{A}}(N, \kappa_\sigma^{\mathsf{SBH}}).$$

Note that

$$\begin{aligned}
&\mathsf{E}[\mathsf{D}_\sigma^{\mathsf{SBH}}|\tilde{A}] \\
&= T_{\max}\mathsf{P}(T_{\tilde{A}} > T_{\max}) + T_{\max}\mathsf{P}(T_{\tilde{A}} \le T_{\max}, X(T_{\tilde{A}}) \in A^{\mathsf{SBH}}) \\
&\quad + T_{\tilde{A}}\mathsf{P}(T_{\tilde{A}} \le T_{\max}, X(T_{\tilde{A}}) \in A \setminus A^{\mathsf{SBH}}) \\
&= T_{\max}\mathsf{P}(T_{\tilde{A}} > T_{\max}) + T_{\max}\Big(1 - \mathsf{P}(T_{\tilde{A}} > T_{\max})\Big)\frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}} \\
&\quad + T_{\tilde{A}}\mathsf{P}(T_{\tilde{A}} \le T_{\max})\Big(1 - \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\Big) \\
&= T_{\max}\frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}} + \Big(1 - \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\Big)T_{\max}\mathsf{P}(T_{\tilde{A}} > T_{\max}) \\
&\quad + \Big(1 - \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\Big)T_{\tilde{A}}\mathsf{P}(T_{\tilde{A}} \le T_{\max}) \\
&= T_{\max}\frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}} + \Big(1 - \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\Big)\mathsf{E}[\min(T_{\tilde{A}}, T_{\max})|\tilde{A}].
\end{aligned}$$

Hence

$$\begin{aligned}
&\mathsf{D}_\sigma^{\mathsf{SBH}}(p) \\
&= T_{\max}\frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\sum_{\tilde{A}\subseteq V}\nu_{\tilde{A}}(N, \kappa_\sigma^{\mathsf{SBH}}) \\
&\quad + \Big(1 - \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\Big)\sum_{\tilde{A}\subseteq V}\mathsf{E}[\min(T_{\tilde{A}}, T_{\max})|\tilde{A}]\Big)\nu_{\tilde{A}}(N, \kappa_\sigma^{\mathsf{SBH}}) \\
&= T_{\max}\frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}} + \Big(1 - \frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}}\Big)\mathsf{T}(\kappa_\sigma^{\mathsf{SBH}}).
\end{aligned}$$

∎

Proof of Corollary 1: For the average server load, it is enough to show that

$$s_\sigma^{\mathsf{SBH}}(p) \ge s_\sigma^{\mathsf{SNR}}(p) \ge s(p). \quad (19)$$

Recall that $s_\sigma^{\mathsf{SNR}}(p) = s((1-\sigma)p),\ \tilde{A} = A^{\mathsf{SBH}} \cup (A \setminus A^{\mathsf{SBH}})$, and $\mathsf{E}[A \setminus A^{\mathsf{SBH}}] = (1-\sigma)pN$. By conditioning on $\tilde{A}$, we have

$$1 - s_\sigma^{\mathsf{SBH}}(p) = \sum_{\tilde{A}\subseteq V}\mathsf{P}(E_1)\nu_{\tilde{A}}$$
$$1 - s_\sigma^{\mathsf{SNR}}(p) = \sum_{\tilde{A}\subseteq V}\mathsf{P}(E_2)\nu_{\tilde{A}}$$

where

$$E_1 = \{\text{event such that } T_{\tilde{A}} < T_{\max} \text{ and } X(T_{\tilde{A}}) \in A \setminus A^{\mathsf{SBH}}\}$$
$$E_2 = \{\text{event such that } T_{A\setminus A^{\mathsf{SBH}}} < T_{\max}\}.$$

Note that $E_1 \subseteq E_2$. Hence

$$\mathsf{P}(T_{\tilde{A}} < T_{\max},\ X(T_{\tilde{A}}) \in A \setminus A^{\mathsf{SBH}}) \le \mathsf{P}(T_{A\setminus A^{\mathsf{SBH}}} < T_{\max}).$$

Therefore

$$1 - s_\sigma^{\mathsf{SBH}}(p) \le 1 - s_\sigma^{\mathsf{SNR}}(p).$$

In (10) of Section IV-C, we have shown that $s_\sigma^{\mathsf{SNR}}(p) \ge s(p)$.

Consider the mean query latency time recalling that $\mathsf{D}_\sigma^{\mathsf{SNR}}(p) = \mathsf{T}\big((1-\sigma)p\big)$ and $\tilde{A} = A^{\mathsf{SBH}} \cup A \setminus A^{\mathsf{SBH}}$. Now consider a path $\mathcal{P} = i_0 \to i_1 \to i_2 \ldots \to i_L$ of the random walk generated by a query request until $T_{\max}$, where i_0 is the peer generating the request and i_j is the n^{th} visited node of $\mathcal{P}$. We call j the index of node i_j.

We can consider the query latency time of this path $\mathcal{P}$ under the presence of blackhole peers (or non-resolving peers), denoted by $\mathsf{D}_\sigma^{\mathsf{SBH}}(\mathcal{P}, \tilde{A})$ (or $\mathsf{D}_\sigma^{\mathsf{SNR}}(\mathcal{P}, \tilde{A})$ respectively), when $\tilde{A}$ is given. The average query latency time is given by conditioning $\mathcal{P}$ and $\tilde{A}$; that is

$$\mathsf{D}_\sigma^{\mathsf{SBH}}(p) = \sum_{\tilde{A}\subseteq V}\sum_{\mathcal{P}}\mathsf{D}_\sigma^{\mathsf{SBH}}(\mathcal{P}, \tilde{A})\mathsf{P}(\mathcal{P})\nu_{\tilde{A}}$$
$$\mathsf{D}_\sigma^{\mathsf{SNR}}(p) = \sum_{\tilde{A}\subseteq V}\sum_{\mathcal{P}}\mathsf{D}_\sigma^{\mathsf{SNR}}(\mathcal{P}, \tilde{A})\mathsf{P}(\mathcal{P})\nu_{\tilde{A}}.$$

Let m be the index of a node of path $\mathcal{P}$ such that $i_m \in A \setminus A^{\mathsf{SBH}}$ and $i_j \notin A \setminus A^{\mathsf{SBH}}$ for all $j < m$. If there is no such $m \le L$, then we set $m = \infty$. Then

$$\mathsf{D}_\sigma^{\mathsf{SNR}}(\mathcal{P}, \tilde{A}) = \begin{cases} T_{i_0}(i_m; \mathcal{P}), & \text{if } m \le L \\ T_{\max}, & \text{otherwise} \end{cases}$$

where $T_{i_0}(i_m; \mathcal{P})$ is the hitting time on i_m starting from i_0 on the path $\mathcal{P}$.

Consider $\mathsf{D}_\sigma^{\mathsf{SBH}}(\mathcal{P}, \tilde{A})$ for the same path $\mathcal{P}$. Note that $\mathsf{D}_\sigma^{\mathsf{SBH}}(\mathcal{P}, \tilde{A}) = \mathsf{D}_\sigma^{\mathsf{SNR}}(\mathcal{P}, \tilde{A})$ if $i_j \notin A^{\mathsf{SBH}}$, for all $j < m$. However, if there is $j < m$ such that $i_j \in A^{\mathsf{SBH}}$, then $\mathsf{D}_\sigma^{\mathsf{SBH}}(\mathcal{P}, \tilde{A}) = T_{\max}$ because the query is not forwarded further. Therefore, $\mathsf{D}_\sigma^{\mathsf{SBH}}(\mathcal{P}, \tilde{A}) \ge \mathsf{D}_\sigma^{\mathsf{SNR}}(\mathcal{P}, \tilde{A})$ holds for all $\mathcal{P}$ and $\tilde{A}$, which means $\mathsf{D}_\sigma^{\mathsf{SBH}}(p) \ge \mathsf{D}_\sigma^{\mathsf{SNR}}(p)$. Recall we have shown that $\mathsf{D}(p) \le \mathsf{D}_\sigma^{\mathsf{SNR}}(p)$ in (11) of Section IV-C.

The fact that $s_\sigma^{\mathsf{SNR}}(p) \le s_\sigma^{\mathsf{SBH}}(p)$ and $\mathsf{D}_\sigma^{\mathsf{SNR}}(p) \le \mathsf{D}_\sigma^{\mathsf{SBH}}(p)$ implies that $s(p)$ and $\mathsf{T}(p)$ are convex as in Corollary 4.

Finally, we show that $\rho_\sigma^{\mathsf{SBH}}(p) \le \rho(p) \le \rho_\sigma^{\mathsf{SNR}}(p)$. Since it is shown that $\rho(p) \le \rho_\sigma^{\mathsf{SNR}}(p)$ in (11) of Section IV-C, it is enough to show that $\rho_\sigma^{\mathsf{SBH}}(p) \le \rho(p)$, which is equivalent to $\mathsf{T}(\kappa_\sigma^{\mathsf{SBH}}) \le (1-\sigma)\mathsf{T}(p)$

$$\begin{aligned}
\mathsf{T}(\kappa_\sigma^{\mathsf{SBH}}) &\le \sigma\mathsf{T}(1) + (1-\sigma)\mathsf{T}(p) \\
&\le (1-\sigma)\mathsf{T}(p)
\end{aligned}$$

since $\mathsf{T}(1) = 0$ and $\mathsf{T}(p)$ is convex as in the Corollary 4. ∎

Using the proof of Corollary 1, we can show that $\mathsf{T}(p)$ and $s(p)$ are convex functions as in Corollary 4.

Corollary 4: $\mathsf{T}(p)$ and $s(p)$ (for the fully cooperative scenario with $\sigma = 0$) are convex functions of p.

Proof: Note that $s(0) = 1$. Hence, $s(p)$ is a convex function of p because, for any $p > 0$ and $\sigma > 0$

$$\begin{aligned} s((1-\sigma)p) &\le s_\sigma^{\mathsf{SBH}}(p) \qquad [\because \text{ by (19)}] \\ &= s(0)\frac{\sigma}{\kappa_\sigma^{\mathsf{SBH}}} + s(\kappa_\sigma^{\mathsf{SBH}})\frac{(1-\sigma)p}{\kappa_\sigma^{\mathsf{SBH}}}. \qquad [\because \text{ by (12)}]. \end{aligned}$$

Using the similar argument and $\mathsf{D}(0) = T_{\max}$, it can be shown that $\mathsf{D}(p)$ is convex. Since $\mathsf{T}(p) = \mathsf{D}(p)$ for the case $\sigma = 0$, $\mathsf{T}(p)$ is convex. ∎

Proof of Proposition 3: First we get the average server load by a similar analysis as in the case of blackhole selfish peers. Recall that $\kappa_\sigma^{\mathsf{SNF}} = \sigma(1-p) + p = \kappa_\sigma^{\mathsf{SBH}}$ as in (14). However, the probability that the query is resolved given that the contact set is reached is

$$\frac{p}{\kappa_\sigma^{\mathsf{SNF}}} = \frac{p}{\sigma(1-p)+p},$$

using the same arguments in the proof of Proposition 2. Hence, (18) becomes

$$1 - s_\sigma^{\mathsf{SNF}}(p) = \frac{p}{\kappa_\sigma^{\mathsf{SNF}}}\Big(1 - s(\kappa_\sigma^{\mathsf{SNF}})\Big).$$

For the average load per peer, the average number of peers who are forwarding a query request or sharing the content is $(1-\sigma)(1-p) + p$ and $\mathsf{T}_\sigma^{\mathsf{SNF}}(p) = \mathsf{T}(\kappa_\sigma^{\mathsf{SBH}})$.

The result of $\mathsf{D}_\sigma^{\mathsf{SNF}}(p)$ can be obtained similarly as in Proposition 2 (here, $\tilde{A} = A \cup A^{\mathsf{SNF}}$). ∎

Proof of Proposition 5: We will show (a), (b), and (c) in terms of the probability that a query is resolved by the server, i.e.,

$$\begin{aligned} &(\mathrm{a}') \quad s_{\sigma_1,\beta_1}^{\mathsf{PNR}}(p) \le s_{\sigma_2,\beta_2}^{\mathsf{PNR}}(p)\ (\le s_{\sigma_1\beta_1}^{\mathsf{SNR}}(p)) \\ &(\mathrm{b}') \quad s_{\sigma_1,\beta_1}^{\mathsf{PNF}}(p) \ge s_{\sigma_2,\beta_2}^{\mathsf{PNF}}(p)\ (\ge s_{\sigma_1\beta_1}^{\mathsf{SNF}}(p)) \\ &(\mathrm{c}') \quad s_{\sigma_1,\beta_1}^{\mathsf{PBH}}(p) \ge s_{\sigma_2,\beta_2}^{\mathsf{PBH}}(p)\ (\ge s_{\sigma_1\beta_1}^{\mathsf{SBH}}(p)). \end{aligned}$$

Let $\mathcal{P} = i_1 \to i_2 \ldots \to i_L$ be the random walk generated by a query request (excluding the peer generating the request), and $Y_n \in \{H, T\}$ be the random coin to decide whether peer i_n possess the query content or not, i.e., $\mathsf{P}(Y_n = H) = p$. Similarly, we use $Z_n \in \{H, T\}$ for i_n being selfish when entering the system, and $W_n \in \{H, T\}$ for i_n acting selfishly when handling the query request. By considering "virtual" walks even after a query request is resolved (or non-forwarded), we assume that the random walk continues until time-to-live $T_{\max}$. Hence, $\mathsf{P}(\mathcal{P})$ is independent of parameters σ, β and types of selfish behaviors, and only dependent on the underlying p2p graph and holding times at peers. Using this notation, we have

$$s(p) = \sum_{\mathcal{P}} \mathsf{P}(\mathcal{P}) \cdot \mathsf{P}(\text{A query is not resolved} \mid \mathcal{P}).$$

Therefore, to prove parts (a), (b), and (c), it suffices to study whether $\mathsf{P}(\text{A query is not resolved} \mid \mathcal{P})$ decreases or increases when σ, β change.

For part (a), i.e., non-resolving peers, we let B_n denote the event that $Y_n = T$ or $Y_n = Z_n = W_n = H$. Using this notation, we have

$$\begin{aligned} \mathsf{P}(\text{A query is not resolved} \mid \mathcal{P}) &= \mathsf{P}\Big(\bigcap_{n=1}^{L} B_n \mid \mathcal{P}\Big) \\ &= \prod_{i \in R(\mathcal{P})} \mathsf{P}\Big(\bigcap_{n \in S_i(\mathcal{P})} B_n \mid \mathcal{P}\Big) \end{aligned} \qquad (20)$$

where $R(\mathcal{P})$ and $S_i(\mathcal{P})$ denote the set of peers appearing in $\mathcal{P}$ and the set of indexes for peer $i \in R(\mathcal{P})$ in $\mathcal{P}$, respectively, i.e.,

$$R(\mathcal{P}) = \{i_1, i_2, \ldots, i_L\} \quad S_i(\mathcal{P}) = \{n : i_n = i\}.$$

For example, if $\mathcal{P} = a \to b \to c \to b \to a \to d$, then $R(\mathcal{P}) = \{a, b, c, d\}$ and $S_a(\mathcal{P}) = \{1, 5\}$, $S_b(\mathcal{P}) = \{2, 4\}$, $S_c(\mathcal{P}) = \{3\}$, $S_d(\mathcal{P}) = \{6\}$. The second equality in (20) is from the independence between $\bigcap_{n \in S_j(\mathcal{P})} B_n$ and $\bigcap_{n \in S_k(\mathcal{P})} B_n$ if $j \neq k$. Furthermore, we have

$$\begin{aligned} &\mathsf{P}(\text{A query is not resolved} \mid \mathcal{P}) \\ &\quad = \prod_{i \in R(\mathcal{P})} \mathsf{P}\Big(\bigcap_{n \in S_i(\mathcal{P})} B_n \mid \mathcal{P}\Big) \\ &\quad = \prod_{i \in R(\mathcal{P})} \mathsf{P}\big(Y_n = T, \forall n \in S_i(\mathcal{P}) \mid \mathcal{P}\big) \\ &\qquad + \mathsf{P}\big(Y_n = Z_n = W_n = H, \forall n \in S_i(\mathcal{P}) \mid \mathcal{P}\big) \\ &\quad = \prod_{i \in R(\mathcal{P})} 1 - p + p\sigma\beta^{|S_i(\mathcal{P})|} \end{aligned} \qquad (21)$$

where we use the fact that $Y_{n_1} = Y_{n_2}$ and $Z_{n_1} = Z_{n_2}$ (with probability 1) for $n_1, n_2 \in S_i(\mathcal{P})$. Equation (21) increases as β increases and $\sigma\beta$ is fixed.

For part (b), i.e., non-forwarding peers, note that

$$\begin{aligned} &\mathsf{P}(\text{A query is resolved} \mid \mathcal{P}) \\ &\quad = \sum_{n=1}^{L} \mathsf{P}(\text{A query is resolved at the } n\text{th peer } i_n \text{ of } \mathcal{P} \mid \mathcal{P}) \\ &\quad = \sum_{n=1}^{L} \mathsf{P}(E_n) \cdot \mathsf{P}(\text{A query is resolved at } i_n \mid E_n, \mathcal{P}) \end{aligned}$$

where we let E_n be the event that the random walk reaches i_n through $\mathcal{P}$, i.e., a query is neither resolved nor non-forwarded until the $(n-1)$th peer of $\mathcal{P}$. One can check that

$$\begin{aligned} &\mathsf{P}(\text{A query is resolved at } i_n \mid E_n, \mathcal{P}) \\ &\quad = \begin{cases} p, & \text{if } i_n \text{ does not appearin } \mathcal{P}_n \\ 0, & \text{otherwise} \end{cases} \\ &\mathsf{P}(E_n) \\ &\quad = \prod_{i \in R(\mathcal{P}_n)} (1-p)(1 - \sigma + \sigma(1-\beta)^{|S_i(\mathcal{P}_n)|}) \end{aligned}$$

where $\mathcal{P}_n = i_1 \to \cdots \to i_{n-1}$, i.e., the first subpath of length $n-1$ in $\mathcal{P}$. The part (b) of Proposition 5 follows by observing that $\mathsf{P}(E_n)$ increases as β increases by Lemma 2.

For part (c), i.e., blackhole peers, we use a similar strategy to that used for the part (b) by using analogous definitions of E_n, $\mathcal{P}_n$. In this case

$$\begin{aligned}&\mathsf{P}(\text{A query is resolved at } i_n \mid E_n, \mathcal{P})\\ &\quad= \begin{cases}(1-\sigma\beta)p, & \text{if } i_n \text{ does not appear in } \mathcal{P}_n\\ 0, & \text{otherwise}\end{cases}\\ &\mathsf{P}(E_n)\\ &\quad= \prod_{i\in R(\mathcal{P}_n)} (1-p)(1-\sigma+\sigma(1-\beta)^{|S_i(\mathcal{P}_n)|}).\end{aligned}$$

As before, this establishes part (c) of Proposition 5. ∎

Lemma 2: The function $1-\sigma+\sigma(1-\beta)^k$ is nondecreasing in β if $k \geq 1$, when $\sigma\beta$ is a constant.

Proof: Using $\sigma\beta = c$ for some constant c, we have

$$1-\sigma+\sigma(1-\beta)^k = 1-\frac{c}{\beta}+\frac{c(1-\beta)^k}{\beta} := b(\beta).$$

When $k = 1$, $b(\beta) = 1-\sigma\beta$, which is a constant since $\sigma\beta$ is fixed. Hence, it is enough to show that for $k > 1$, $b(\beta)$ is increasing; $b'(\beta) > 0$. Taking a derivative, we have

$$b'(\beta) = \frac{c}{\beta^2}\Big(1-(1-\beta)^{k-1}\big((k-1)\beta+1\big)\Big).$$

In order to show that $b'(\beta) > 0$, we will show that

$$a(x) = (1-x)^m(mx+1) \leq 1$$

for $m(=k-1) > 0$. Taking a logarithm on $a(x)$, $\log a(x) = m\log(1-x)+\log(mx+1)$. Hence

$$\big(\log a(x)\big)' = -\frac{m}{1-x}+\frac{m}{mx+1}.$$

Note that both $-\frac{m}{1-x}$ and $\frac{m}{mx+1}$ are decreasing functions of x. Therefore, $\big(\log a(x)\big)'$ is decreasing with x. Moreover, $\big(\log a(x)\big)' = 0$ at $x = 0$. These two observations conclude that $\log a(x)$ takes maximum at $x = 0$. The maximum of $\log a(x) = 0$, i.e., $a(x)$ takes it maximum, 1, at $x = 0$. That is, $b'(\beta) \geq 0$. ∎

Proof of Proposition 6: As in the proof of Proposition 5, recall path $\mathcal{P} = i_1 \to \cdots \to i_{L(\mathcal{P})}$ and its subpath, $\mathcal{P}_n = i_1 \to \cdots \to i_{n-1}$ and the random variables Y_n, Z_n, W_n. Now define $R_n(\mathcal{P}) = \{i_1, i_2, .., i_{n-1}\}$ and $S_{i,n}(\mathcal{P}) = \{m \mid i_m = i,\ m < n\}$. Recall that $H_\sigma^{\mathsf{SNR}}(p)$ is the average number of hops of the random walk search for a single query request until the query request terminates (i.e., the query request is resolved for non-resolving peers) and that $\rho_\sigma^{\mathsf{SNR}}(p)$ is proportional to $H_\sigma^{\mathsf{SNR}}(p)$.

For part (a), by conditioning on path $\mathcal{P}$

$$\begin{aligned}&H_\sigma^{\mathsf{SNR}}(p)\\ &\quad= \sum_{\mathcal{P}} \mathsf{P}(\mathcal{P}) \sum_{m=1}^{L(\mathcal{P})} m\cdot \mathsf{P}(\text{A query terminates at } i_m \mid \mathcal{P})\\ &\quad= \sum_{\mathcal{P}} \mathsf{P}(\mathcal{P}) \sum_{m=1}^{L(\mathcal{P})} m\cdot \mathsf{P}(E_m)\mathsf{P}(F_m \mid E_m, \mathcal{P})\end{aligned}$$

where E_m is the event that the random walk reaches i_m through $\mathcal{P}$ and F_m is the event that the random walk terminates (i.e., is resolved) at i_m. Then, it can be checked that

$$\begin{aligned}\mathsf{P}(E_m) &= \prod_{i\in R_m(\mathcal{P}} \Big((1-p)+p\sigma\beta^{|S_{i,n}(\mathcal{P})|}\Big)\\ &= \prod_{i\in R_m(\mathcal{P}} \Big((1-p)+pc\beta^{|S_{i,n}(\mathcal{P})|-1}\Big)\\ \mathsf{P}(F_m \mid E_m, \mathcal{P}) &= p(1-\sigma)+p\sigma(1-\beta)\\ &= 1-p\sigma\beta.\end{aligned}$$

Note that $\mathsf{P}(F_m \mid E_m, \mathcal{P})$ is a constant (since $\sigma\beta$ is a constant) and $\mathsf{P}(E_m)$ is increasing with β. Therefore, the load per peer defined by $\rho_\sigma^{\mathsf{SNR}}(p)$ is a increasing function of β, which results in (a).

For part (b) and part (c), with the same argument, we have

$$\begin{aligned}\mathsf{P}(E_m) &= \prod_{i\in R_m(\mathcal{P}} (1-p)\Big(1-\sigma+\sigma(1-\beta)^{|S_{i,m}(\mathcal{P})|}\Big)\\ \mathsf{P}(F_m|E_m,\mathcal{P}) &= p+(1-p)\sigma\beta.\end{aligned}$$

Note that $\mathsf{P}(F_m \mid E_m, \mathcal{P})$ is a constant and $\mathsf{P}(E_m)$ is increasing, which is shown in the proof of part (b) of Proposition 5. ∎

References

[1] C. Huang, J. Li, and K. W. Ross, "Can Internet video-on-demand be profitable?," in *Proc. ACM SIGCOMM*, 2007, pp. 133–144.
[2] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On next-generation telco-managed P2P TV architectures," in *Proc. USENIX IPTPS*, 2008, p. 5.
[3] Y. Chen *et al.*, "Towards capacity and profit optimization of video-on-demand services in a peer-assisted IPTV platform," *Multimedia Syst.*, vol. 15, no. 1, pp. 19–32, 2009.
[4] J. Cho and Y. Yi, "On the payoff mechanisms in peer-assisted services with multiple content providers: Rationality and fairness," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 731–744, Jun. 2014.
[5] V. Misra, S. Inoannidis, A. Chaintreau, and L. Maasoulié, "Incentivizing peer-assisted services: A fluid Shapley value approach," in *Proc. ACM SIGMETRICS*, 2010, pp. 215–226.
[6] Y. Jin, Y. Yi, G. Kesidis, F. Kocak, and J. Shin, "Hybrid client-server and peer-to-peer caching systems with selfish peers," in *Proc. IEEE INFOCOM*, 2013, pp. 1744–1752.
[7] C. Zhao, X. Lin, and C. Wu, "The streaming capacity of sparsely-connected p2p systems with distributed control," in *Proc. IEEE INFOCOM*, 2011, pp. 1449–1457.
[8] C. Zhao, J. Zhao, X. Lin, and C. Wu, "Capacity of P2P on-demand streaming with simple, robust and decentralized control," in *Proc. IEEE INFOCOM*, 2013, pp. 2697–2705.
[9] D. Ciullo, V. Martina, M. Garetto, and E. Leonardi, "How much can large-scale video-on-demand benefit from users' cooperation?," in *Proc. IEEE INFOCOM*, 2013, pp. 2724–2732.
[10] S. Ioannidis and P. Marbach, "On the design of hybrid peer-to-peer systems," in *Proc. ACM SIGMETRICS*, Annapolis, MD, USA, Jun. 2008, pp. 157–168.
[11] E. Adar and B. A. Huberman, "Free riding on Gnutella," *First Monday*, vol. 5, no. 10, 2000.
[12] D. Huges, G. Coulson, and J. Walkerdine, "Free riding on Gnutella revisted: The bell tolls?," *IEEE Distrib. Syst. Online*, vol. 6, no. 6, 2005.
[13] M. R. Rahman, "A survey of incentive mechanism in peer-to-peer systems," 2009 [Online]. Available: http://www.cs.uwaterloo.ca/research/tr/2009/CS-2009-22.pdf
[14] C. Li, B. Yu, and K. Sycara, "An incentive mechanism for message relaying in unstructured peer-to-peer systems," in *Proc. AAMAS*, 2007, pp. 271–278.
[15] J. Kleinberg and P. Raghavan, "Query incentive networks," in *Proc. ACM FOCS*, 2005, pp. 132–141.

[16] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proc. ACM Conf. Electron. Commerce*, 2004, pp. 102–111.
[17] R. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.
[18] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bull. AMS*, vol. 43, no. 4, pp. 439–561, 2006.
[19] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks," in *Proc. IEEE INFOCOM*, 2004, pp. 120–130.
[20] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge, "Incentives for sharing in peer-to-peer networks," in *Proc. 2nd Int. Workshop Electron. Commerce*, 2001, pp. 75–87.
[21] S. Marti and H. Garica-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Comput. Netw.*, vol. 50, no. 4, pp. 472–484, 2006.
[22] S. Félegyházi, J. Hubaux, and L. Buttyán, "Nash equilibria of packet forwarding strategies in wireless ad hoc network," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 463–476, May 2006.
[23] V. Srinavasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, 2003, pp. 808–817.
[24] L. Buttyan and J. P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mobile Netw. Appl.*, vol. 8, no. 5, pp. 579–592, 2003.
[25] B. Yang, T. Condie, S. Kamvar, and H. Garcia-Molina, "Non-cooperation in competitive P2P networks," in *Proc. IEEE ICDCS*, 2005, pp. 91–100.
[26] S. Ioannidis, "Scalable and reliable searching in unstructured peer-to-peer systems," Ph.D. dissertation, University of Toronto, Toronto, ON, Canada, 2009.
[27] C. Law and K.-Y. Su, "Distributed construction of random expander networks," in *Proc. IEEE INFOCOM*, 2003, pp. 2133–2143.
[28] D. B. West, *Introduction to Graph Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

Youngmi Jin (S'00–M'05) received the B.S. and M.S. degrees in mathematics from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1991 and 1993, respectively, and the M.S. and Ph.D. degrees in electrical engineering from The Pennsylvania State University, University Park, PA, USA, in 2000 and 2005, respectively.

She worked with the University of Pennsylvania, Philadelphia, PA, USA, and KT, Sogang University, and KAIST in Korea. She is currently with KDDI R&D Labs, Saitama, Japan, as a Research Engineer. Her research interests include virtualization, cloud computing, social networks, Internet economics, and wireless networks.

George Kesidis received the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, CA, USA, in 1990 and 1992, respectively.

Following 8 years as a Professor of electrical and computer engineering with the University of Waterloo, Waterloo, ON, Canada, he has been a Professor of computer science and engineering and electrical engineering with The Pennsylvania State University, University Park, PA, USA, since 2000. His research interests include different aspects of computer networking and cyber security, more recently including issues of energy efficiency and network economics.

Jinwoo Shin received the B.S. degree in computer science and mathematics from Seoul National University, Seoul, Korea, in 2001, and the Ph.D. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2010.

After spending 3 years with the Georgia Institute of Technology, Atlanta, GA, USA, and the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, he joined the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2013 as an Assistant Professor.

Dr. Shin received the Best MIT CS Doctoral Thesis (George M. Sprowls) Award in 2010, the Best Publication Award in INFORMS Applied Probability in 2013, and two other conference Best Paper awards.

Fatih Kocak received the B.S. and M.S. degrees in electrical and electronic engineering from Bilkent University, Ankara, Turkey, in 2007 and 2010, respectively, and the Ph.D. degree in electrical engineering from The Pennsylvania State University, University Park, PA, USA, in 2014.

He is currently with Maxim Integrated, San Francisco, CA, USA, working on image processing solutions on touchscreens of android devices. His research interests include machine learning applications in various areas, including network intrusion detection, and implications of network neutrality.

Yung Yi (S'04–M'06) received the B.S. and M.S. degrees in computer science and engineering from Seoul National University, Seoul, Korea, in 1997 and 1999, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2006.

Now, he is an Associate Professor with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His current research interests include the design and analysis of computer networking systems.

Dr. Yi was the recipient of two Best Paper awards at IEEE SECON 2013 and ACM MobiHoc 2013.