Analysis of TTL-Based Consistency in Unstructured Peer-to-Peer Networks

Xueyan Tang, *Member*, *IEEE*, Jianliang Xu, *Senior Member*, *IEEE*, and Wang-Chien Lee, *Member*, *IEEE*

Abstract—Consistency maintenance is important to the sharing of dynamic contents in peer-to-peer (P2P) networks. The TTL-based mechanism is a natural choice for maintaining freshness in P2P content sharing. This paper investigates TTL-based consistency maintenance in unstructured P2P networks. In this approach, each replica is assigned an expiration time beyond which the replica stops serving new requests unless it is validated. While TTL-based consistency is widely explored in many client-server applications, there has been no study on TTL-based consistency in P2P networks. Our main contribution is an analytical model that studies the search performance and the freshness of P2P content sharing under TTL-based consistency. Due to the random nature of request routing, P2P networks are fundamentally different from most existing TTL-based systems in that every node with a valid replica has the potential to serve any other node. We identify and discuss the factors that affect the performance of P2P content sharing under TTL-based consistency. Our results indicate a tradeoff between search performance and freshness: the search cost decreases sublinearly with decreasing freshness of P2P content sharing. We also compare two types of unstructured P2P networks and find that clustered P2P networks improve the freshness of content sharing over flat P2P networks under TTL-based consistency.

Index Terms—Unstructured P2P network, TTL-based consistency, replication, consistency maintenance, content distribution.

1 Introduction

Peer-to-peer (P2P) networks are one of the most prevalent applications in today's Internet. These networks interconnect millions of nodes to form an ad hoc overlay network and share contents through searching and replication. With the proliferation of new P2P applications that share dynamic contents (e.g., directory services and surveillance data), there is an increasing need to maintain freshness in P2P content sharing [1]. Freshness and search performance are two important measures for the sharing of dynamic contents. Search performance concerns how fast the users locate and obtain copies of requested contents while freshness concerns how "old" the acquired copies are with respect to the authoritative contents. Consistency maintenance in P2P content sharing has started receiving attention in recent years [2], [3], [4].

In this paper, we investigate TTL-based consistency maintenance in P2P networks. In this approach, each replica is assigned a time-to-live (TTL) value.[1] When its TTL expires, a replica is not allowed to serve new requests unless it is validated. TTL-based consistency is suitable for P2P networks due to a number of reasons. First, TTL-based consistency does not require the content owner to keep track of the replica locations. Therefore, it is resilient to node joins and leaves and thus suitable for highly dynamic systems like P2P networks. Second, since each replica determines its validity autonomously, validations of expired replicas can be performed with either the content owners or other valid replicas [6]. This coincides in spirit with the nature of P2P networks in the sense that each replica has the potential to serve any other replica in consistency maintenance. Meanwhile, it also helps to offload the content owner, which is advantageous in large-scale networks. Third, the TTL-based scheme guarantees that the staleness of content sharing is time-bounded. It offers the flexibility to use different TTL values to cater for different consistency requirements that may vary with the nature of shared contents and user tolerance (e.g., in distributed file systems, the TTL value of a file is often smaller than that of a directory [7]; in the Domain Name System, NS records tend to have larger TTL values than A records [8]). The smaller the TTL value, the higher the consistency level. A TTL value of 0 is equivalent to the strong consistency guarantee. Last, TTL-based consistency is readily supported by the HTTP protocol [6] that prevails in P2P file downloads [9]. Therefore, the TTL-based mechanism is a natural choice for maintaining freshness in P2P content sharing.

While TTL-based consistency is widely explored in many client-server applications (e.g., distributed file systems [10], web caching [6], and the Domain Name System [8]), there has been no study on TTL-based consistency in P2P networks. In this paper, we focus on unstructured P2P networks (e.g., Gnutella [11] and FastTrack/KaZaA [9]) since they are more widely deployed than structured ones. Unstructured P2P networks are fundamentally different from the aforementioned systems that employ TTL-based

1. We remark that the term TTL in this paper is a different one from that used in controlled flooding over P2P networks [5]. In controlled flooding, a TTL is associated with a message to control the number of hops that the message can be propagated. The TTL we discuss here is associated with a replica to control how long the replica can be used without validating its freshness.

- *X. Tang is with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798. E-mail: asxytang@ntu.edu.sg.*
- *J. Xu is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: xujl@comp.hkbu.edu.hk.*
- *W.-C. Lee is with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802. E-mail: wlee@cse.psu.edu.*

Manuscript received 29 July 2007; revised 7 Feb. 2008; accepted 11 Feb. 2008; published online 29 Feb. 2008.
Recommended for acceptance by K. Hwang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-07-0257.
Digital Object Identifier no. 10.1109/TPDS.2008.44.

consistency. The replicas in the aforementioned systems are often organized into a static hierarchy [6], [8], [12]. A replica, on expiration, always contacts its parent in the hierarchy for validation. Meanwhile, the clients, located at the bottom level of the hierarchy, do not serve each other. Due to network dynamics, such a hierarchy is difficult to maintain in P2P networks. In an unstructured P2P network, there is no coupling between the nodes and the locations of contents. Thus, request routing is done by random search so that every node with a valid replica has the potential to serve any other node. Therefore, existing studies on TTL-based consistency are not applicable to P2P content sharing. There is a need to understand the performance implications of adopting TTL-based consistency in P2P networks.

The main contribution of this paper is an analytical model that studies the search performance and the freshness of P2P content sharing under TTL-based consistency. The analysis is validated by a wide range of simulation experiments. Our results indicate a tradeoff between search performance and freshness: the search cost decreases sublinearly with decreasing freshness of P2P content sharing, and the TTL value may serve as a tuning knob between these two measures. We also compare two types of unstructured P2P networks and find that clustered P2P networks improve the freshness of content sharing over flat P2P networks under TTL-based consistency.

The rest of this paper is organized as follows: Section 2 summarizes the related work. Section 3 applies TTL-based consistency to unstructured P2P networks. Section 4 develops an analytical model and investigates a variety of performance measures. The experimental setup and results are discussed in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

There are several studies on replication strategies in P2P networks. Lv et al. [5] experimentally showed that replication improves the search performance of unstructured P2P networks. To optimize network-wide search performance given limited storage capacity, more replicas are preferred for more frequently accessed objects. Cohen and Shenker [13] proved that the search time and traffic under random walk search is minimized when the number of replicas for each object is proportional to the square root of its query rate. Tewari and Kleinrock [14] showed that under controlled flooding search, the search traffic is minimized under the same square-root replica distribution, whereas the search time is minimized when the number of replicas for each object is linearly proportional to its query rate. Different from [13] and [14], Kangasharju et al. [15] developed a logarithmic replication distribution to maximize the content availability under intermittent connectivity. However, none of the above work has considered keeping the replicas consistent with the authoritative contents.

In general, there are two classes of methods to maintain consistency: push-based and pull-based. In push-based methods, the content owners keep track of the replica locations and send invalidation messages or updated contents to the replicas whenever the contents are modified. In contrast, pull-based methods are replica-driven. The replicas, when considered outdated, are validated before serving new requests. Chen et al. [3] proposed a push-based method to maintain strong consistency in structured P2P networks where the content placement is tightly coupled with the network topology [16], [17]. They built a hierarchical replica-partition-tree to track all replica locations and propagated updates to the replicas through application-level multicast. However, such a tree is difficult to construct and maintain in unstructured P2P networks. An alternative method that does not need to keep track of replica locations is to propagate updates through flooding [4]. Nevertheless, flooding has high communication overhead. Datta et al. [2] replaced flooding with rumor spreading to reduce the overhead of update propagation. This method offers probabilistic rather than deterministic guarantees in consistency maintenance. Liu et al. [4] also proposed an adaptive pull-based method in which all nodes poll the content owners to check for consistency. This not only requires the nodes to record the owners for all objects but also turns the content owners into bottlenecks. While most of the above mechanisms require new implementations, TTL-based consistency is readily available in the HTTP protocol [6] that is widely used for P2P content sharing. So far, there has been no study on TTL-based consistency in P2P networks despite its advantages discussed in the Introduction.

Some work has been done on TTL-based consistency in web caching. Dilley [18] investigated the impact of TTL-based consistency on the response time of web caches. Jung et al. [19] analyzed the hit rate of a single TTL-based cache. Iyer et al. [20] organized a set of cooperative web caches into a structured P2P network to facilitate searching for cached contents, while Xiao et al. [21] proposed a P2P web caching management scheme to make the browsers as well as their proxy share cached contents. In these systems, the participating caches are conceptually equivalent to a single aggregate cache. Hence, the consistency maintenance is the same as that of a single TTL-based cache. The performance of a multilevel TTL-based cache hierarchy was analyzed in [6] and [22]. In our earlier work, we explored the optimization of request routing and replica placement under TTL-based consistency [12], [23]. However, all the above work assumed a hierarchical client-server architecture in which the requests are routed along static tree structures for resolution. In contrast, request routing in unstructured P2P networks is done by random search. As a result, every node with a valid replica has the potential to serve any other node. Therefore, the above studies are not applicable to P2P content sharing. TTL-based consistency in unstructured P2P networks requires new models to analyze.

3 Applying TTL-Based Consistency in Unstructured P2P Networks

3.1 Basic Definitions

We term any generic contents provided by the nodes in an unstructured P2P network for the other nodes to download as *objects*. Each object is hosted by one or more nodes in the network called its *authoritative nodes*. These nodes are managed by the content owner who makes updates to the object contents. The remaining nodes are called *nonauthoritative nodes*. The nonauthoritative nodes are much larger in

number than the authoritative nodes. To speed up accesses, copies of an object may also be stored at nonauthoritative nodes. These copies are called *replicas*.

To maintain freshness for content sharing, each replica is associated with an *expiration time* that controls how long the replica might be used. If a query arrives before the expiration time, the replica is considered *valid* and can be used by the requesting node for downloading. The time interval between the creation of a replica and its expiration time is called the *lifetime*. Note that the update behavior of the content owner at the authoritative nodes is independent of TTL-based consistency.

In the following, we elaborate the operation of TTL-based consistency in two types of unstructured P2P networks: flat and clustered networks.

3.2 Flat P2P Network

In a flat P2P network, all nodes play the same role. Each node maintains a set of "neighbors" to form an overlay network. The nodes do not keep any index of the objects available at the other nodes. When a node requests for an object that is not available locally, it has to search for the object by sending queries to the others. A node receiving a query responds with its object copy if it is an authoritative node or a nonauthoritative node having a valid replica. Otherwise, the receiving node forwards the query to some other nodes. Due to the lack of information on possible locations of objects, the search in unstructured P2P networks is usually "blind." Two widely used search methods are flooding [14] and random walk [24]. In flooding, the query is forwarded to all neighbors of the receiving node, whereas in random walk, the query is forwarded to a randomly selected neighbor.

On resolving a query, the downloaded object copy creates a replica at the requesting node. An expiration time is assigned to the replica. Under TTL-based consistency, if the replica is created by downloading from an authoritative node, its expiration time is assigned the value $t + T$, where t is the current time, and T is the *TTL value*. Otherwise, if the replica is created by downloading from a nonauthoritative node, its expiration time is set to that of the valid replica at the nonauthoritative node. Due to the inheritance of expiration time, the lifetime of the new replica cannot exceed T. As a result, an object copy can at most be used for a time period equal to the TTL value T after leaving the authoritative nodes. Therefore, it guarantees a consistency level at which the staleness of content sharing is bounded by an interval T.

3.3 Clustered P2P Network

In a clustered P2P network, the nodes are classified into *supernodes* and *ordinary nodes* [9]. The supernodes connect among themselves to form an overlay network just like the nodes in a flat P2P network. Each ordinary node, on the other hand, connects to a supernode and provides the information of its local objects to the supernode. A supernode and all ordinary nodes connected to it form a *cluster*. Each supernode maintains an index of all the objects available in its local cluster.

When an ordinary node requests for an object that is not available locally, it sends a query to its supernode. If the supernode locates a valid object copy in the local cluster, it sends the object location to the requesting node straightaway. Otherwise, the supernode performs a search among the other supernodes like that in a flat P2P network and then replies to the requesting node with the object location. After that, the requesting node downloads the object and creates a replica. The assignment methodology of an expiration time to the new replica is the same as that discussed in flat P2P networks. The requesting node also informs its supernode of the new replica (together with the expiration time) and the supernode updates the index. Again, TTL-based consistency ensures that the staleness of content sharing is bounded by an interval T.

4 Performance Analysis

Now, we analyze the performance of TTL-based consistency in unstructured P2P networks. We start with flat P2P networks and then extend the results to clustered P2P networks.

4.1 General Model

Consider the sharing of a single object in a flat P2P network consisting of N nodes. Suppose there are N_a authoritative nodes of the object. The remaining $N - N_a$ nodes are nonauthoritative nodes. Let $\gamma = N_a/N$ be the ratio of authoritative nodes in the network, then the portion of nonauthoritative nodes in the network is $1 - \gamma$. Suppose each nonauthoritative node requests for the object at a rate of λ. Then, the total query rate for the object is $(1 - \gamma)N\lambda$.

Assume the TTL-based scheme is used for consistency maintenance. Let T be the TTL value. For ease of presentation, we say that a nonauthoritative node is *valid (invalid)* if it has (does not have) a valid copy of the object. If a requesting node is valid at the time of query, the query constitutes a *hit*. Otherwise, if the requesting node is invalid, the query is a *miss* and has to be resolved through searching the other nodes. On resolving the query, the requesting node becomes valid and remains valid until the acquired object copy expires.[2] We say that a valid node has remainder lifetime t if the object copy at the node has remainder lifetime t.

To analyze performance, we consider a discretized approximation of the system by classifying the nonauthoritative nodes into $k+1$ categories, where k is an integer. Category 0 contains invalid nodes. Let $\Delta t = T/k$. The rest k categories are defined by dividing all possible remainder lifetimes $(0, T]$ into k disjoint ranges $(0, \Delta t]$, $(\Delta t, 2 \cdot \Delta t], (2 \cdot \Delta t, 3 \cdot \Delta t], \ldots, ((k-1) \cdot \Delta t, k \cdot \Delta t = T]$. For each $1 \le i \le k$, category i contains valid nodes having remainder lifetimes in the range $((i-1) \cdot \Delta t, i \cdot \Delta t]$. It is intuitive that the accuracy of approximation improves as k increases (or equivalently, $\Delta t \to 0$). We start by analyzing the discretized approximation and then take $k \to \infty$ to derive the system performance.

Let p_i $(0 \le i \le k)$ be the portion of nodes in category i, among all nodes in the network. It is obvious that $\sum_{i=0}^{k} p_i = 1 - \gamma$. For each $1 \le i \le k$, the nodes in category i transfer to category $(i-1)$ as time elapses. Since any node with a remainder lifetime in $((i-1) \cdot \Delta t, i \cdot \Delta t]$ at present

2. To focus on the performance implications of TTL-based consistency, we assume that the storage capacity of the nodes is large enough to hold any object copy acquired until it expires.

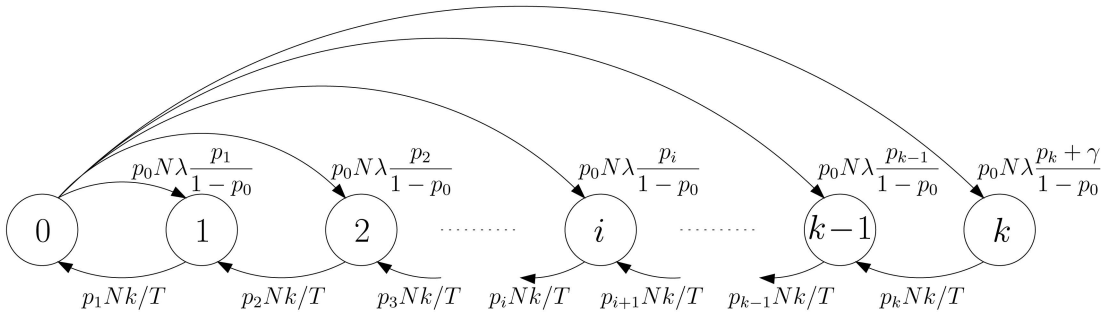


Fig. 1. State transition diagram.

would have a remainder lifetime in $((i-2)\cdot\Delta t,(i-1)\cdot\Delta t]$ after Δt time, the transition rate from category i to $(i-1)$ is given by $p_iN/\Delta t = p_iNk/T$. On the other hand, when an invalid node requests for the object, it has to search for a valid object copy. On acquiring the valid copy, the node transfers to one of the categories 1 to k. So, the total transition rate from category 0 to the other categories is $p_0N\lambda$. Note that the valid copy can be obtained from either an authoritative node or a valid nonauthoritative node. In the former case, the remainder lifetime of the requesting node would be set to T. In the latter case, the remainder lifetime of the requesting node would be set to that of the valid nonauthoritative node. In the absence of information about which other nodes may better be able to resolve queries, the performance of blind search in unstructured P2P networks is well captured by random probes [5], [13]. In random probes, the requesting node repeatedly draws a node uniformly at random and probes it for the requested object until a valid object copy is found. Hence, each authoritative node and valid nonauthoritative node is equally likely to be the node eventually resolving the query. Note that the total portion of authoritative nodes and valid nonauthoritative nodes in the network is $p_1+p_2+\cdots+p_k+\gamma=1-p_0$, where γ is the ratio of authoritative nodes. Therefore, the probabilities that the requesting node transfers to categories $1,2,\ldots,k-1,k$ are given by $p_1/(1-p_0),p_2/(1-p_0),\ldots,p_{k-1}/(1-p_0),(p_k+\gamma)/(1-p_0)$, respectively. Fig. 1 summarizes the transition rates of the nodes between categories.

At the steady state, flow must be conserved in the sense that the flow into each category must equal the flow out from the category [25]. Therefore, the inbound transition rate of each category is the same as the outbound transition rate. Thus, we have the following equilibrium equations:

$$\begin{cases} p_0N\lambda\cdot\frac{p_k+\gamma}{1-p_0}=p_kN\cdot\frac{k}{T} \\ p_kN\cdot\frac{k}{T}+p_0N\lambda\cdot\frac{p_{k-1}}{1-p_0}=p_{k-1}N\cdot\frac{k}{T} \\ \cdots\quad\cdots \\ p_{i+1}N\cdot\frac{k}{T}+p_0N\lambda\cdot\frac{p_i}{1-p_0}=p_iN\cdot\frac{k}{T} \\ \cdots\quad\cdots \\ p_2N\cdot\frac{k}{T}+p_0N\lambda\cdot\frac{p_1}{1-p_0}=p_1N\cdot\frac{k}{T} \\ p_1N\cdot\frac{k}{T}=p_0N\lambda. \end{cases}$$

It follows that

$$\begin{cases} p_k=\frac{p_0\lambda T\gamma}{k-kp_0-p_0\lambda T} \\ p_k=p_{k-1}\cdot\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right) \\ \cdots\quad\cdots \\ p_{i+1}=p_i\cdot\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right) \\ \cdots\quad\cdots \\ p_2=p_1\cdot\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right) \\ p_1=p_0\lambda\cdot\frac{T}{k}. \end{cases} \tag{1}$$

The first equation in (1) implies

$$\frac{p_0\lambda T\gamma}{k(1-p_0)}=p_k\cdot\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right). \tag{2}$$

Integrating the last k equations in (1), we have

$$p_k=p_0\lambda\cdot\frac{T}{k}\cdot\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right)^{k-1}. \tag{3}$$

Combining (2) and (3), we obtain

$$\frac{\gamma}{1-p_0}=\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right)^{k}.$$

When $k\to\infty$, it is easy to establish that

$$\lim_{k\to\infty}\left(1-\frac{p_0\lambda T}{k(1-p_0)}\right)^{k}=e^{-\frac{p_0\lambda T}{1-p_0}}. \tag{4}$$

Therefore, p_0 satisfies

$$\frac{\gamma}{1-p_0}=e^{-\frac{p_0\lambda T}{1-p_0}}=e^{\lambda T\left(1-\frac{1}{1-p_0}\right)}. \tag{5}$$

Equation (5) implies that p_0 depends on γ (the ratio of authoritative nodes in the network) and the product λT (the expected number of queries issued per node in an interval T).

Let us examine two special cases. At one extreme, if $T=0$ (i.e., the system enforces strong consistency and does not allow sharing of any stale contents), $p_0=1-\gamma$. This implies no replica is considered valid and every query results in a download from the authoritative nodes. At the other extreme, if $T\to\infty$ (i.e., the system does not have any consistency requirement), $p_0\to 0$. That is, each node would eventually possess a replica (in the absence of storage constraint) and all replicas are valid.

Although it is difficult to present a closed form solution of p_0 for the general case, it is easy to solve p_0 numerically. Note that the left-hand side of (5) increases monotonically with p_0, while the right-hand side decreases monotonically with p_0. When $p_0 = 0$, the left-hand side equals γ, which is less than the right-hand side (i.e., 1). When $p_0 = 1 - \gamma e^{-\lambda T}$, the left-hand side equals $e^{\lambda T}$, which is greater than the right-hand side. Therefore, p_0 must lie in the interval $(0, 1 - \gamma e^{-\lambda T})$. Thus, p_0 can be efficiently computed by the Newton or bisection method. In the following, we shall derive various performance metrics as a function of p_0.

4.2 Search Performance

The performance metric we consider for searching is the number of nodes probed by a requesting node to find the requested object. This is known as the *search size* [13]. We take the search size as a primary measure because the search costs in terms of network traffic and query processing load are both proportional to search size. It is also a good indicator of search time: the search time of random walk is linearly proportional to search size and the search time of controlled flooding is logarithmically proportional to search size [14].

Note that the total number of authoritative nodes and valid nonauthoritative nodes is given by $(1 - p_0)N$. Hence, in random probes, an average of $N/((1 - p_0)N)$ nodes need to be probed to resolve a query miss. Therefore, the average search size of query misses is

$$\frac{1}{1 - p_0}.$$

Among the nonauthoritative nodes, a portion $p_0/(1 - \gamma)$ of them are invalid. Thus, a portion $p_0/(1 - \gamma)$ of all queries are misses and the remaining portion $(1 - \gamma - p_0)/(1 - \gamma)$ of queries are hits. Since the search sizes of hits are 0, the average search size of all queries is

$$\frac{p_0}{(1 - p_0) \cdot (1 - \gamma)}.$$

Recall that p_0 depends on γ and λT. Therefore, the search size is affected by the ratio of authoritative nodes in the network and the expected number of queries issued per node in an interval T (the TTL value).

4.3 Distribution of Remainder Lifetime

We now compute the remainder lifetimes of object copies at the nonauthoritative nodes. They will be used to compute the freshness of content sharing in Section 4.4.

Let $F(t)$ be the cumulative distribution function of remainder lifetimes, i.e., the probability of the nodes being valid and having remainder lifetimes less than t.

Suppose $t/\Delta t$ is an integer. Then,

$$F(t) = p_1 + p_2 + \cdots + p_{\frac{t}{\Delta t}}.$$

It follows from (1) that

$$\begin{aligned} F(t) &= p_0\lambda \cdot \frac{T}{k}\left(1 + \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right) + \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right)^2 \right. \\ &\qquad \left. + \cdots + \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right)^{\frac{t}{\Delta t}-1}\right) \\ &= p_0\lambda \cdot \frac{T}{k} \cdot \frac{1 - \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right)^{\frac{t}{\Delta t}}}{1 - \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right)} \\ &= (1 - p_0) \cdot \left(1 - \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right)^{\frac{t}{\Delta t}}\right) \\ &= (1 - p_0) \cdot \left(1 - \left(1 - \frac{p_0\lambda T}{k(1-p_0)}\right)^{k \cdot \frac{t}{T}}\right). \end{aligned}$$

Letting $k \to \infty$, it follows from (4) that

$$\begin{aligned} F(t) &= (1 - p_0) \cdot \left(1 - e^{-\frac{p_0\lambda T}{1-p_0} \cdot \frac{t}{T}}\right) \\ &= (1 - p_0) \cdot \left(1 - e^{-\frac{p_0\lambda t}{1-p_0}}\right) \\ &= 1 - p_0 - (1 - p_0) \cdot e^{-\frac{p_0\lambda t}{1-p_0}}. \end{aligned}$$

Based on (5), it is easy to verify that $F(0) = 0$, and

$$\begin{aligned} F(T) &= 1 - p_0 - (1 - p_0) \cdot e^{-\frac{p_0\lambda T}{1-p_0}} \\ &= 1 - p_0 - (1 - p_0) \cdot \frac{\gamma}{1 - p_0} \\ &= 1 - p_0 - \gamma \end{aligned}$$

(since a portion $1 - p_0 - \gamma$ of the nodes are nonauthoritative but valid).

Taking the first-order derivative of $F(t)$, we obtain the probability density function $f(t)$ of remainder lifetimes as

$$\begin{aligned} f(t) = \frac{\mathrm{d}F(t)}{\mathrm{d}t} &= -(1 - p_0) \cdot \left(-\frac{p_0\lambda}{1 - p_0}\right) \cdot e^{-\frac{p_0\lambda t}{1-p_0}} \\ &= p_0\lambda \cdot e^{-\frac{p_0\lambda t}{1-p_0}}. \end{aligned}$$

Equation (5) implies that for any t,

$$\left(\frac{\gamma}{1 - p_0}\right)^{\frac{t}{T}} = e^{-\frac{p_0\lambda t}{1-p_0}}.$$

Therefore,

$$f(t) = p_0\lambda \cdot \left(\frac{\gamma}{1 - p_0}\right)^{\frac{t}{T}}.$$

It is seen that $f(t)$ decreases exponentially with t, i.e., there are more nodes with short remainder lifetimes than with long remainder lifetimes in the network.

4.4 Freshness of Content Sharing

We define the *age* of an object copy as the elapsed time since it was first obtained from an authoritative node [6]. Age is a good indicator of the freshness of an object copy. In general, the lower the age, the fresher the object copy. An age of 0 implies the object copy must be up-to-date. Under TTL-based consistency, the maximum age of object copies is

bounded by T. Besides this worst-case measure, it is also important to examine an average measure. In this section, we compute the average age of object copies obtained by users upon queries.

Note that the age of an object copy equals T minus its remainder lifetime. We first compute the average remainder lifetime of object copies returned upon query misses. Remember that each authoritative node and valid nonauthoritative node is equally likely to be the node resolving a query. These nodes constitute a portion $1-p_0$ of all nodes in the network. With probability $\gamma/(1-p_0)$, a query miss is resolved by an authoritative node. In this case, the remainder lifetime of the object copy acquired is T. With probability $(1-p_0-\gamma)/(1-p_0)$, a query miss is resolved by a valid nonauthoritative node. In this case, the remainder lifetime of the object copy acquired follows the probability density function derived in Section 4.3. Therefore, the average remainder lifetime of object copies returned upon query misses is given by

$$\mathcal{R}_{\text{miss}} = \frac{\gamma}{1-p_0}\cdot T + \frac{1-p_0-\gamma}{1-p_0}\cdot\frac{\int_0^T f(t)\cdot t\,\mathrm{d}t}{\int_0^T f(t)\,\mathrm{d}t}.$$

Note that

$$\int_0^T f(t)\,\mathrm{d}t = F(T)-F(0) = 1-p_0-\gamma, \tag{6}$$

and

$$\begin{aligned}
&\int_0^T f(t)\cdot t\,\mathrm{d}t\\
&= \int_0^T p_0\lambda\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\cdot t\,\mathrm{d}t\\
&= \int_0^T p_0\lambda\cdot\left(-\frac{1-p_0}{p_0\lambda}\right)\cdot t\,\mathrm{d}e^{-\frac{p_0\lambda t}{1-p_0}}\\
&= -(1-p_0)t\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\Big|_0^T - \int_0^T -(1-p_0)\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\,\mathrm{d}t\\
&= -(1-p_0)T\cdot e^{-\frac{p_0\lambda T}{1-p_0}} + (1-p_0)\cdot\left(-\frac{1-p_0}{p_0\lambda}\right)\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\Big|_0^T\\
&= -(1-p_0)T\cdot e^{-\frac{p_0\lambda T}{1-p_0}} - \frac{(1-p_0)^2}{p_0\lambda}\cdot e^{-\frac{p_0\lambda T}{1-p_0}} + \frac{(1-p_0)^2}{p_0\lambda}\\
&= -(1-p_0)T\cdot\frac{\gamma}{1-p_0} - \frac{(1-p_0)^2}{p_0\lambda}\cdot\frac{\gamma}{1-p_0} + \frac{(1-p_0)^2}{p_0\lambda}\\
&= -\gamma T - \frac{(1-p_0)\gamma}{p_0\lambda} + \frac{(1-p_0)^2}{p_0\lambda}\\
&= -\gamma T + \frac{(1-p_0)(1-p_0-\gamma)}{p_0\lambda}.
\end{aligned} \tag{7}$$

Thus,

$$\begin{aligned}
\mathcal{R}_{\text{miss}} &= \frac{\gamma T}{1-p_0} + \frac{\int_0^T f(t)\cdot t\,\mathrm{d}t}{1-p_0}\\
&= \frac{\gamma T}{1-p_0} - \frac{\gamma T}{1-p_0} + \frac{1-p_0-\gamma}{p_0\lambda}\\
&= \frac{1-p_0-\gamma}{p_0\lambda}.
\end{aligned}$$

Therefore, the average age of object copies obtained upon query misses is

$$\mathcal{A}_{\text{miss}} = T - \mathcal{R}_{\text{miss}} = T - \frac{1-p_0-\gamma}{p_0\lambda}.$$

Now, let us take the query hits into account as well. Recall that if a query miss is resolved by an authoritative node, the requesting node obtains a lifetime T. The expected number of subsequent hits within the lifetime is λT. The average remainder lifetime upon these hits is $(1/2)\cdot T$. Similarly, if a query miss is resolved by a valid nonauthoritative node and obtains a lifetime t, the expected number of subsequent hits within the lifetime is λt. The average remainder lifetime upon these hits is $(1/2)\cdot t$. Therefore, the average remainder lifetime of object copies obtained by users over all queries is

$$\mathcal{R} = \frac{\gamma\cdot\left(T+\lambda T\cdot\frac{1}{2}T\right) + \int_0^T f(t)\left(t+\lambda t\cdot\frac{1}{2}t\right)\mathrm{d}t}{\gamma\cdot(1+\lambda T) + \int_0^T f(t)(1+\lambda t)\,\mathrm{d}t}. \tag{8}$$

Note that

$$\begin{aligned}
&\int_0^T f(t)\cdot t^2\,\mathrm{d}t\\
&= \int_0^T p_0\lambda\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\cdot t^2\,\mathrm{d}t\\
&= \int_0^T p_0\lambda\cdot\left(-\frac{1-p_0}{p_0\lambda}\right)\cdot t^2\,\mathrm{d}e^{-\frac{p_0\lambda t}{1-p_0}}\\
&= -(1-p_0)t^2\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\Big|_0^T - \int_0^T -(1-p_0)\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\,\mathrm{d}t^2\\
&= -(1-p_0)T^2\cdot e^{-\frac{p_0\lambda T}{1-p_0}} + \int_0^T 2(1-p_0)\cdot e^{-\frac{p_0\lambda t}{1-p_0}}\cdot t\,\mathrm{d}t\\
&= -(1-p_0)T^2\cdot\frac{\gamma}{1-p_0} + \frac{2(1-p_0)}{p_0\lambda}\cdot\int_0^T f(t)\cdot t\,\mathrm{d}t\\
&= -\gamma T^2 + \frac{2(1-p_0)}{p_0\lambda}\cdot\left(-\gamma T - \frac{(1-p_0)\gamma}{p_0\lambda} + \frac{(1-p_0)^2}{p_0\lambda}\right)\\
&= -\gamma T^2 - \frac{2(1-p_0)\gamma T}{p_0\lambda} - \frac{2(1-p_0)^2\gamma}{p_0^2\lambda^2} + \frac{2(1-p_0)^3}{p_0^2\lambda^2}\\
&= -\gamma T^2 - \frac{2(1-p_0)\gamma T}{p_0\lambda} + \frac{2(1-p_0)^2(1-p_0-\gamma)}{p_0^2\lambda^2}.
\end{aligned} \tag{9}$$

Integrating (6), (7), and (9) into (8), we have

$$\begin{aligned}
\mathcal{R} &= \frac{\frac{(1-p_0)(1-p_0-\gamma)}{p_0\lambda} - \frac{(1-p_0)\gamma T}{p_0} + \frac{(1-p_0)^2(1-p_0-\gamma)}{p_0^2\lambda}}{1-p_0-\frac{(1-p_0)\gamma}{p_0}+\frac{(1-p_0)^2}{p_0}}\\
&= \frac{\frac{1-p_0-\gamma}{p_0\lambda} - \frac{\gamma T}{p_0} + \frac{(1-p_0)(1-p_0-\gamma)}{p_0^2\lambda}}{1-\frac{\gamma}{p_0}+\frac{1-p_0}{p_0}}\\
&= \frac{\frac{1-p_0-\gamma}{p_0\lambda} - \gamma T}{1-\gamma}.
\end{aligned}$$

Therefore, the average age of object copies obtained by users over all queries is

$$\mathcal{A} = T - \mathcal{R} = \frac{T - \frac{1-p_0-\gamma}{p_0\lambda}}{1-\gamma}. \tag{10}$$

Comparing $\mathcal{A}$ with $\mathcal{A}_{\text{miss}}$, it is interesting to find that they differ by a constant factor $(1-\gamma)$ that is solely determined by the ratio of authoritative nodes in the network.

4.5 Extensions to Clustered P2P Networks

Now, we extend the analysis to clustered P2P networks. Suppose in a clustered P2P network of N nodes, each cluster contains n nodes. Then, there are N/n clusters. We continue to denote by γ, λ, and T, respectively, the ratio of authoritative nodes, the query rate per node, and the TTL value.

In a clustered P2P network, a requesting node always downloads the object from the local cluster if possible. It goes through its supernode to search for the object in the other clusters only when there is no valid copy in the local cluster. Thus, in a cluster without any authoritative node, all valid replicas, if any, must have the same age. On the other hand, if a cluster contains an authoritative node, the corresponding supernode always responds to queries with the location of the authoritative node. Therefore, a clustered P2P network is conceptually similar to a flat P2P network in the sense that each cluster is equivalent to a single node that initiates queries at a cluster-aggregate rate $\lambda^* = n\lambda$. Just like the nodes in a flat P2P network, the clusters in a clustered P2P network can be classified into authoritative and nonauthoritative ones depending on whether a cluster contains an authoritative node. Recall that there are γN authoritative nodes. Suppose they are spread out in $C \leq \gamma N$ clusters. Then, the ratio of authoritative clusters to all clusters is given by

$$\gamma^* = \frac{C}{N/n} = \frac{nC}{N}.$$

Let p_0^* be the portion of clusters that are nonauthoritative but contain a valid object copy. Following (5) in Section 4.1, p_0^* satisfies

$$\frac{\gamma^*}{1-p_0^*} = e^{\lambda^* T\left(1-\frac{1}{1-p_0^*}\right)}.$$

Following the result in Section 4.2, the average number of nonlocal supernodes to be probed for each query originating from a nonauthoritative cluster is

$$\frac{p_0^*}{(1-p_0^*)\cdot(1-\gamma^*)}.$$

Note that roughly a portion γ^* of the nonauthoritative nodes are in the authoritative clusters. Queries originating from these nodes do not need to probe any nonlocal supernode. Taking into account the probe of the local supernode, the average search size of all queries is bounded by

$$\frac{p_0^*}{(1-p_0^*)\cdot(1-\gamma^*)}\cdot(1-\gamma^*)+1 = \frac{1}{(1-p_0^*)}.$$

Following (10) in Section 4.4, the average age of object copies obtained upon the queries originating from nonauthoritative clusters is

$$\mathcal{A}^*_{\text{nonauth}} = \frac{T-\frac{1-p_0^*-\gamma^*}{p_0^*\lambda^*}}{1-\gamma^*}.$$

For each nonauthoritative node in an authoritative cluster, a query miss always acquires an object copy of age 0 and is expected to be followed by λT query hits. The average age of the object copy upon these hits is $(1/2)\cdot T$. Thus, the average age of object copies obtained by users over all queries is

$$\begin{aligned}\mathcal{A}^* &= \mathcal{A}^*_{\text{nonauth}}\cdot(1-\gamma^*)+\frac{\lambda T\cdot\frac{1}{2}T}{1+\lambda T}\cdot\gamma^*\\ &= T-\frac{1-p_0^*-\gamma^*}{p_0^*\lambda^*}+\frac{\lambda T^2}{2+2\lambda T}\cdot\gamma^*.\end{aligned}$$

5 Experimental Evaluation

5.1 Experimental Setup

We have conducted simulation experiments to study the performance implications of TTL-based consistency in unstructured P2P networks. The experimental results are also compared with the analytical results derived in Section 4.

In the experiments, we used uniform random graphs for the overlay network topologies in flat P2P networks.[3] Given a number of N nodes, a random graph was generated by inserting an edge between each pair of nodes with a connecting probability l. The larger the value l, the higher the connectivity of the graph. We have experimented with many network topologies of different sizes and connecting probabilities. The results all showed similar performance trends. Due to space limitations, we shall report only the informative results obtained with a network of 100,000 nodes and a connecting probability of 0.00012 (so the average node degree is 12).

We simulated the sharing of one object. Given a ratio $0<\gamma<1$, a portion of γN nodes were randomly selected in the network to be the authoritative nodes of the object. The default γ was set at 0.0001 (i.e., there were 10 authoritative nodes). The remaining (nonauthoritative) nodes each requested the object following Poisson arrivals at a rate λ. TTL-based consistency was employed with a TTL value T. We performed experiments over a wide range of settings for these parameters to study their impacts. The search mechanism we simulated was single-walker random walk. That is, if a node receiving a query did not have a valid object copy, it forwarded the query to one of its neighbors selected at random (to avoid the ping-pong effect, the neighbor through which this node received the query was excluded from selection unless it is the only neighbor of the receiving node, i.e., the degree of the receiving node is 1). Each simulation run started with no valid replica available at any nonauthoritative node and simulated a total of 100 million query arrivals in the entire network.

5.2 Impact of TTL Value

We first study a flat P2P network. Fig. 2a shows the average search size of queries as a function of T when λ was set at one query per time unit. We also simulated the network

3. It is known that for flat P2P networks, uniformly random overlay topologies yield the best search performance [5].

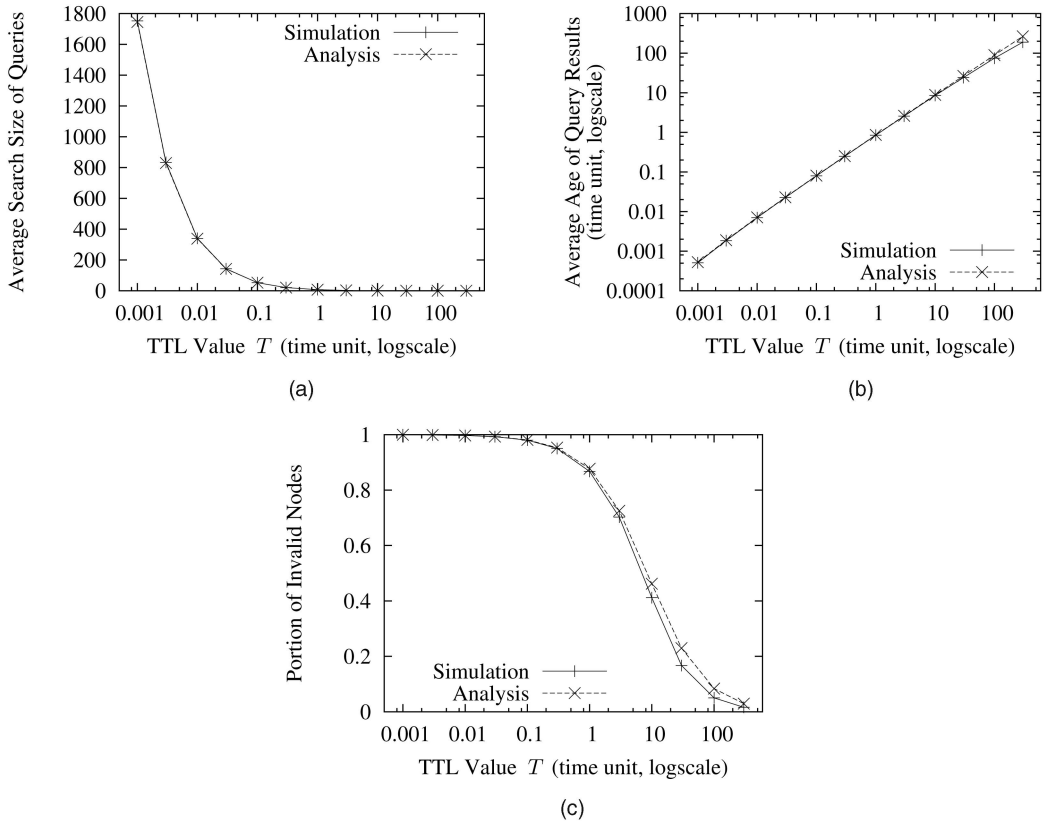


Fig. 2. Performance for different TTL values T. (a) Search size. (b) Age of object copies obtained upon queries. (c) Portion of invalid nodes.

without replication so that all queries have to be resolved by authoritative nodes. The resultant search size, 9,473, is much larger than those presented in Fig. 2a. This implies replication can significantly improve search performance even in the presence of TTL-based consistency maintenance.

Fig. 2b shows that the average age of object copies acquired upon queries increases with T. We also plot in Fig. 2c the portion of invalid nodes in the network. This portion was calculated in the simulation as follows: We sampled the P2P network once every 0.1 time unit to count how many nonauthoritative nodes did not have valid object copies. The counts were then averaged at the end of the simulation.

When T is small, the replicas expire rapidly. As shown in Fig. 2c, most nodes are invalid. Thus, a large number of nodes need to be probed to locate a valid object copy, resulting in large search size (see the left side of Fig. 2a). In this case, many queries are eventually resolved by authoritative nodes and obtain object copies of age 0. Therefore, as seen from Fig. 2b, the average age of object copies obtained upon queries is low. With increasing T, the portion of invalid nodes decreases as the replicas are valid for longer periods of time. As a result, the search size also reduces (see Fig. 2a). Meanwhile, at a lower consistency level, a larger portion of queries are resolved by replicas. Hence, the average age of object copies obtained upon queries increases (see Fig. 2b) since the object copies acquired from replicas have higher ages than those from authoritative nodes. Combining the results in Figs. 2a and 2b, we plot in Fig. 3 the relationship between the search size and the age of object copies acquired upon queries as T varies. Figs. 3a and 3b show the relationship for different ranges of age values. These results indicate a tradeoff between search performance and freshness: the search cost decreases sublinearly with decreasing freshness of P2P content sharing, and the TTL value may serve as a tuning knob between these two measures.

Besides the simulation results, we also show in Fig. 2 the results derived from the analysis in Section 4. It is seen that the simulated and analytical results match well for a wide range of T values. There is, however, some difference between the two results at high T values, which will be explained later.

5.3 Impact of Query Rate

Figs. 4a and 4c show the search size and the portion of invalid nodes, respectively, for different query rates λ when T was set at 1 time unit. When λ is low, only a few replicas exist in the network. Thus, the portion of invalid nodes is high and the search size is large. With increasing λ, more replicas are created in the network due to queries. As a result, the portion of invalid nodes decreases and the search size reduces.

It is also seen that varying the query rate λ under a fixed TTL value T is symmetric to varying T under a fixed λ. In fact, the coordinates of the curves in Figs. 4a and 4c are numerically almost the same as those in Figs. 2a and 2c. This verifies our analytical observation (in Section 4.2) that p_0 and, hence, the search size depends only on the product λT rather than the individual λ and T values. Therefore, fixing T (or λ) at different values would simply shift the

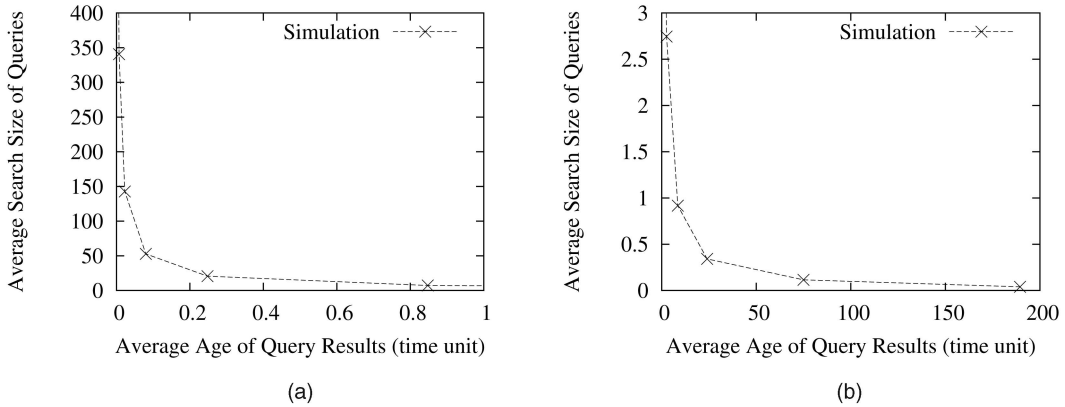


Fig. 3. Relationship between search size and age of query results as TTL value T varies. (a) Search size versus age of query results. (b) Search size versus age of query results.

curves horizontally in Figs. 4a and 4c (or Figs. 2a and 2c) but would not change the shape of the curves.

Similar to Fig. 2, Fig. 4 also shows that the simulated and analytical results match closely for a wide range of λ values, but there is some difference between them at high λ values. To investigate, we plot in Fig. 5 the distribution of remainder lifetimes for valid replicas. The simulation results in Fig. 5a were again calculated by sampling the simulated P2P network once every 0.1 time unit. At each sample, we recorded the remainder lifetimes of all valid replicas. The distribution statistics were computed by aggregating the recorded values over all samples. The analytical results in Fig. 5b are the probability density function derived in Section 4.3. As seen from Fig. 5, the analytical results agree with the simulated results quite well when $\lambda T \leq 10$. When λT is large (i.e., interquery times per node are negligible compared to T), however, the curves of simulated results flatten out. A close look at the simulated network reveals that in this case, the system does not tend to converge to a steady state. Instead, it evolves in a cyclic manner. Fig. 6 shows, over a sample interval of $5T$ in the simulation, the instantaneous portion of invalid nodes in the network as time elapses. Starting from an initial state without any replica, all nonauthoritative nodes issue queries in a short interval relative to T. Thus, they all obtain object copies of nearly 0 age. As a result, all replicas

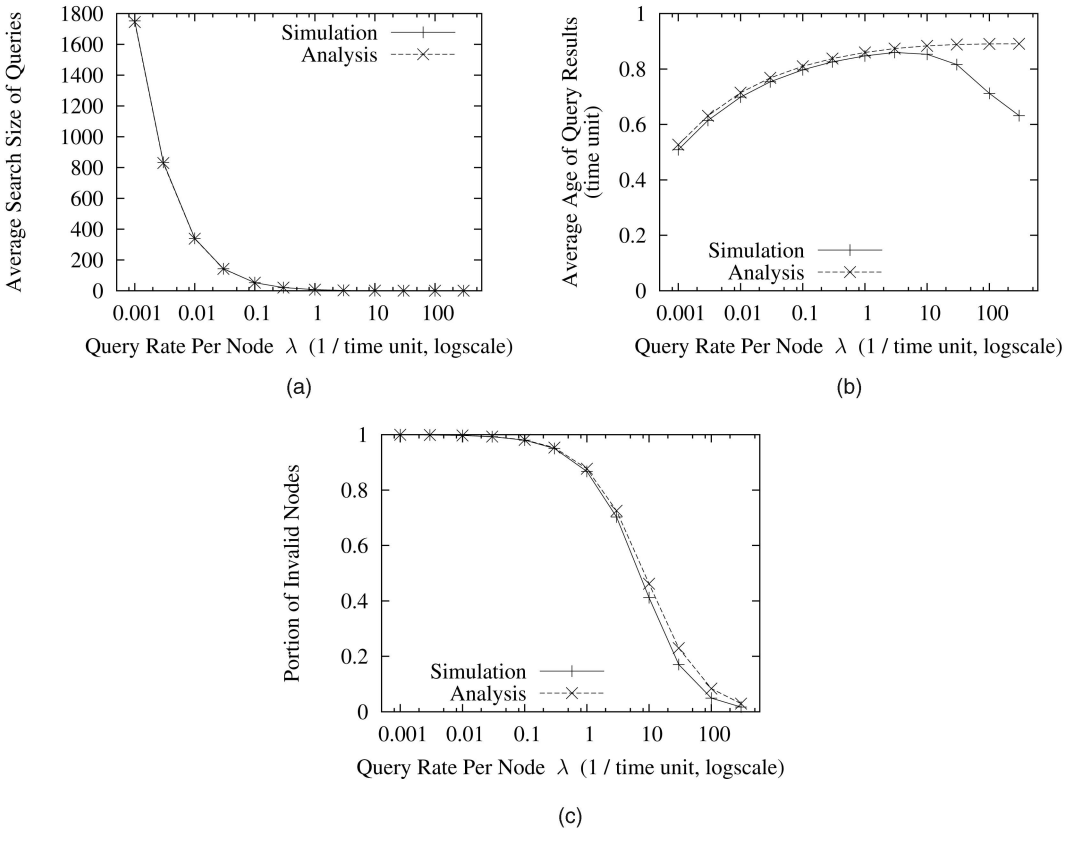


Fig. 4. Performance for different query rates λ. (a) Search size. (b) Age of object copies obtained upon queries. (c) Portion of invalid nodes.

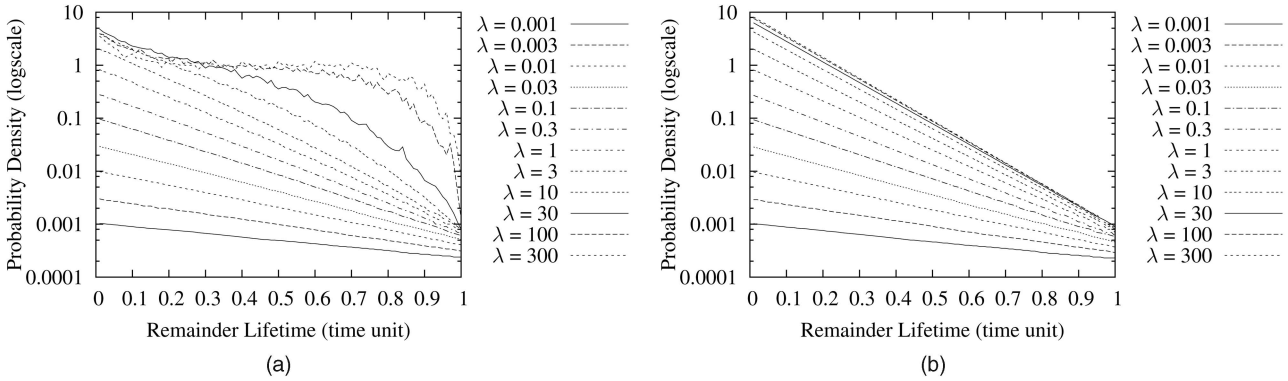


Fig. 5. Distribution of remainder lifetime versus query rate λ. (a) Simulation results. (b) Analytical results.

created would expire at almost the same time. After expiration, all nodes issue queries again in a short interval and repeat the process. Therefore, the probability density function of remainder lifetimes for valid replicas is close to uniform (see Fig. 5a). Note that the queries generated by a nonauthoritative node during the lifetime of its replica are resolved locally at the node. These queries form the vast majority of all queries when λT is large. Thus, the ages of object copies obtained upon queries approximate a uniform distribution over $[0, T]$. This explains why the average age of object copies acquired upon queries deviates from analytical results and approaches $(1/2) \cdot T = 1/2$ at high λ values in Fig. 4b.

5.4 Impact of Ratio of Authoritative Nodes

Fig. 7a shows that the search size reduces logarithmically with increasing ratio of authoritative nodes γ (note that the x-axis is in logscale). When γ is small, adding an authoritative node into the network cuts down the search size significantly. However, the performance gain due to each new authoritative node declines with increasing γ.

A larger γ generally increases the chance for queries to be resolved by authoritative nodes, thereby reducing the ages of object copies obtained upon queries. In Fig. 7b, we plot the average age of object copies acquired upon query misses (i.e., $\mathcal{A}_{\text{miss}}$ as defined in Section 4.4) as well as that of object copies returned over all queries (i.e., $\mathcal{A}$ as defined in Section 4.4). Fig. 7b shows that the difference between the two average ages increases with γ. For example, at $\gamma = 0.01$, $\mathcal{A}$ is about 1 percent higher than $\mathcal{A}_{\text{miss}}$, while at $\gamma = 0.1$, $\mathcal{A}$ is about 10 percent higher than $\mathcal{A}_{\text{miss}}$. This confirms our analytical result that $\mathcal{A}$ and $\mathcal{A}_{\text{miss}}$ differ by a constant factor $(1 - \gamma)$.

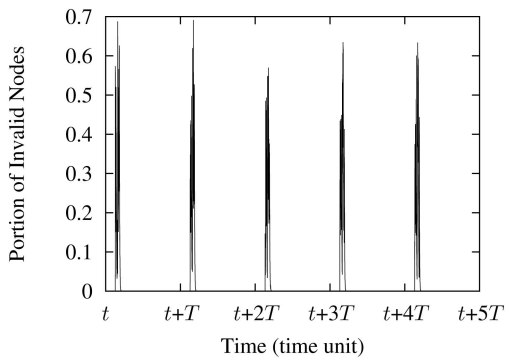


Fig. 6. Portion of invalid nodes over time $(\lambda T = 300)$.

Recall from Section 4.5 that a clustered P2P network is conceptually similar to a flat P2P network in the sense that each cluster is equivalent to a single "aggregate" node. We have conducted experiments for clustered P2P networks with the authoritative nodes spreading out in different numbers of clusters. The results, as shown in Fig. 8, reflect similar performance trends: the average intercluster search cost, i.e., the mean number of nonlocal supernodes to probe for each query, decreases with increasing number of authoritative clusters. Therefore, it is beneficial for the content owner to spread out the authoritative nodes in different clusters in a clustered P2P network.

5.5 Comparison of Flat and Clustered Networks

Finally, we study clustered P2P networks. While it is known that clustered networks can substantially improve search performance over flat networks, our objective here is to compare their freshness of content sharing under TTL-based consistency. In our experiments, the size of a clustered network was set the same as that of a flat network, i.e., 100,000 nodes. Following empirical observations [9], each cluster was assumed to contain 100 nodes. Thus, there were 1,000 clusters and supernodes. The overlay network topology for the supernodes was a random graph with the same average node degree as that of the flat network. In the experiments, the authoritative nodes in the clustered network were deployed in different clusters since it improves the search performance and freshness of content sharing as discussed in Section 5.4. The parameters γ, T, and λ were set the same in both flat and clustered networks.

Fig. 9 shows the average age of object copies obtained upon queries for different T values. For ease of comparison, the average ages are normalized by the T values. As discussed earlier, when T is small, most queries are resolved by authoritative nodes. Thus, the two networks do not differ much in the age of object copies obtained upon queries. As T increases, more queries are resolved by replicas. Fig. 9 shows that the nodes obtain object copies of significantly lower ages in the clustered network than in the flat network. For example, at $T = 3$ and 10, the clustered network results in about 40 percent lower ages than the flat network. This implies that organizing the nodes into clusters helps improve the freshness of content sharing.

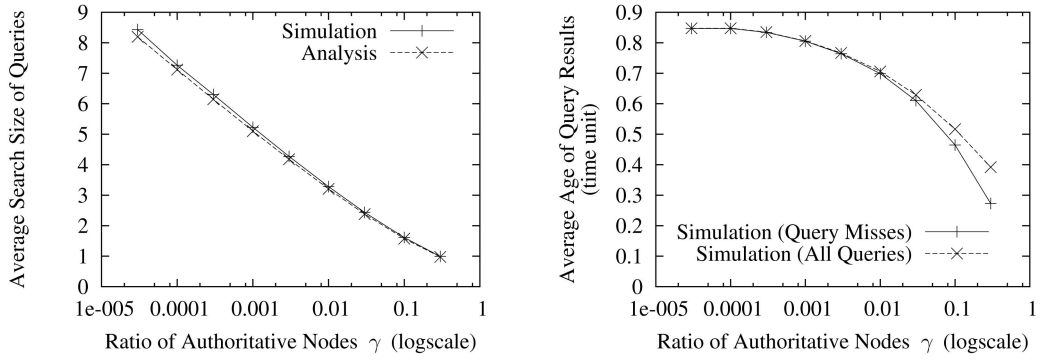


Fig. 7. Performance for different ratios of authoritative nodes γ ($T = 1$ time unit, $\lambda = 1$ query per time unit). (a) Search size. (b) Age of object copies obtained upon queries.

6 Conclusion

TTL-based consistency is a widely used method to maintain the freshness of content sharing in the Internet. In this paper, we have investigated the performance of TTL-based consistency in unstructured P2P networks. We have built an analytical model to study the search performance and freshness of content sharing in both flat and clustered P2P networks. The analysis has been validated by a wide range of simulation experiments. The results show that: 1) replication substantially improves the performance of P2P content sharing even under TTL-based consistency maintenance; 2) there is a tradeoff between search performance and freshness: the search cost decreases sublinearly with decreasing freshness of P2P content sharing; and 3) a clustered P2P network architecture not only improves search performance, but also improves the freshness of content sharing under TTL-based consistency.

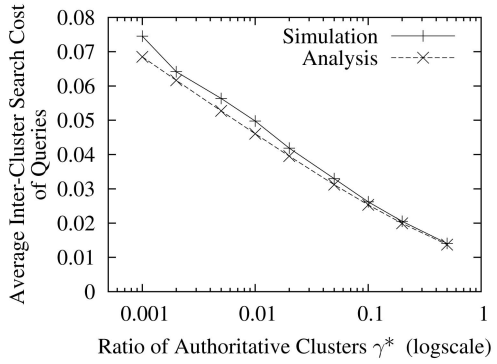


Fig. 8. Intercluster search cost for different ratios of authoritative clusters γ^* (for a clustered network of 1,000 clusters each containing 100 nodes, $T = 1$ time unit, $\lambda = 1$ query per time unit).

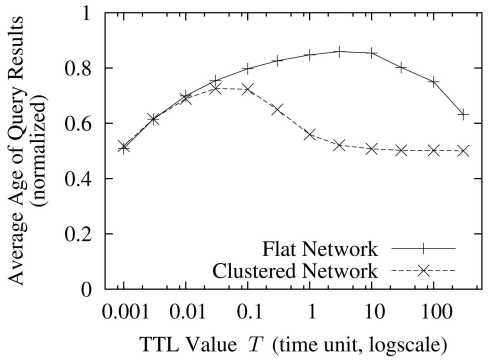


Fig. 9. Age of object copies obtained upon queries (simulation results).

Acknowledgments

The work of Jianliang Xu was supported in part by the Research Grants Council of Hong Kong under Projects HKBU211206 and HKBU211307. The work of Wang-Chien Lee was supported in part by the US National Science Foundation under Grants IIS-0328881, IIS-0534343, and CNS-0626709.

References

[1] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335-371, Dec. 2004.

[2] A. Datta, M. Hauswirth, and K. Aberer, "Updates in Highly Unreliable, Replicated Peer-to-Peer Systems," *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS '03)*, May 2003.

[3] X. Chen, S. Ren, H. Wang, and X. Zhang, "SCOPE: Scalable Consistency Maintenance in Structured P2P Systems," *Proc. IEEE INFOCOM '05*, Mar. 2005.

[4] X. Liu, J. Lan, P. Shenoy, and K. Ramaritham, "Consistency Maintenance in Dynamic Peer-to-Peer Overlay Networks," *Computer Networks*, vol. 50, no. 6, pp. 859-876, Apr. 2006.

[5] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," *Proc. ACM Int'l Conf. Supercomputing (ICS '02)*, June 2002.

[6] E. Cohen and H. Kaplan, "Aging through Cascaded Caches: Performance Issues in the Distribution of Web Content," *Proc. ACM SIGCOMM '01*, pp. 41-53, Aug. 2001.

[7] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, fourth ed., Addison-Wesley, 2005.

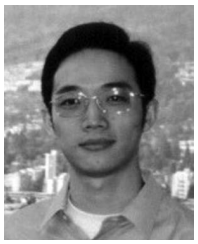
[8] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS Performance and the Effectiveness of Caching," *IEEE/ACM Trans. Networking*, vol. 10, no. 5, pp. 589-603, Oct. 2002.

[9] J. Liang, R. Kumar, and K.W. Ross, "The FastTrack Overlay: A Measurement Study," *Computer Networks*, vol. 50, no. 6, pp. 842-858, Apr. 2006.

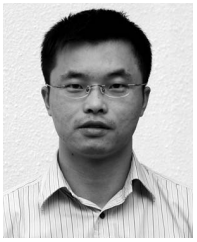
[10] V. Cate, "Alex—A Global File System," *Proc. USENIX File System Workshop*, pp. 1-12, May 1992.

[11] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-Like P2P Systems Scalable," *Proc. ACM SIGCOMM '03*, Aug. 2003.

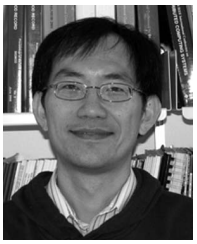
[12] X. Tang and S.T. Chanson, "The Minimal Cost Distribution Tree Problem for Recursive Expiration-Based Consistency Management," *IEEE Trans. Parallel and Distributed Systems,* vol. 15, no. 3, pp. 214-227, Mar. 2004.
[13] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," *Proc. ACM SIGCOMM '02,* Aug. 2002.
[14] S. Tewari and L. Kleinrock, "Proportional Replication in Peer-to-Peer Networks," *Proc. IEEE INFOCOM '06,* Apr. 2006.
[15] J. Kangasharju, K. Ross, and D. Turner, "Adaptive Content Management in Structured P2P Communities," *Proc. Int'l ICST Conf. Scalable Information Systems (INFOSCALE '06),* June 2006.
[16] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. ACM SIGCOMM '01,* pp. 149-160, Aug. 2001.
[17] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 1, Jan. 2004.
[18] J. Dilley, "The Effect of Consistency on Cache Response Time," *IEEE Network,* vol. 14, no. 3, pp. 24-28, May/June 2000.
[19] J. Jung, A.W. Berger, and H. Balakrishnan, "Modeling TTL-Based Internet Caches," *Proc. IEEE INFOCOM '03,* Apr. 2003.
[20] S. Iyer, A. Rowstron, and P. Druschel, "Squirrel: A Decentralized Peer-to-Peer Web Cache," *Proc. ACM Symp. Principles of Distributed Computing (PODC '02),* July 2002.
[21] L. Xiao, X. Zhang, A. Andrzejak, and S. Chen, "Building a Large and Efficient Hybrid Peer-to-Peer Internet Caching System," *IEEE Trans. Knowledge and Data Eng.,* vol. 16, no. 6, pp. 754-769, June 2004.
[22] Y.T. Hou, J. Pan, B. Li, and S. Panwar, "On Expiration-Based Hierarchical Caching Systems," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 1, pp. 134-150, Jan. 2004.
[23] X. Tang and S.T. Chanson, "Analysis of Replica Placement under Expiration-Based Consistency Management," *IEEE Trans. Parallel and Distributed Systems,* vol. 17, no. 11, pp. 1253-1263, Nov. 2006.
[24] C. Gkantsidis, M. Mihail, and A. Saberi, "Random Walks in Peer-to-Peer Networks," *Proc. IEEE INFOCOM '04,* Mar. 2004.
[25] L. Kleinrock, *Queueing Systems, Volume I: Theory.* John Wiley & Sons, 1975.



Xueyan Tang received the BEng degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 1998 and the PhD degree in computer science from the Hong Kong University of Science and Technology in 2003. He is currently an assistant professor in the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include mobile and pervasive computing, wireless sensor networks, Web and Internet, and distributed systems. He has published more than 40 technical papers in the aforementioned areas, mostly in prestigious journals and conference proceedings. He is an editor of a book entitled *Web Content Delivery* published by Springer. He has also served as a program committee member for many international conferences. He is a member of the IEEE.



Jianliang Xu received the BEng degree in computer science and engineering from Zhejiang University, Hangzhou, China, in 1998 and the PhD degree in computer science from the Hong Kong University of Science and Technology in 2002. He is an associate professor in the Department of Computer Science, Hong Kong Baptist University. He was a visiting scholar in the Department of Computer Science and Engineering, Pennsylvania State University, University Park. His research interests include data management, mobile/pervasive computing, wireless sensor networks, and distributed systems. He has published more than 50 technical papers in these areas, most of which appeared in prestigious journals and conference proceedings, including ACM SIGMOD, IEEE ICDE, IEEE INFOCOM, the *IEEE/ACM Transactions on Networking*, the *IEEE Transactions on Parallel and Distributed Systems*, and the *IEEE Transactions on Knowledge and Data Engineering*. He is an editor of a book entitled *Web Content Delivery*, published by Springer, and a coguest editor of the *International Journal of Grid Computing: Theory, Methods and Applications* (Elsevier) for a special issue on scalable information systems. He also serves as the vice chairman of ACM Hong Kong Chapter. He is a senior member of the IEEE.



Wang-Chien Lee received the BS degree from the National Chiao Tung University, Hsinchu, Taiwan, the MS degree from Indiana University, Bloomington, and the PhD degree from the Ohio State University, Columbus. He is an associate professor of computer science and engineering at the Pennsylvania State University, University Park, where he leads the Pervasive Data Access (PDA) Research Group to perform cross-area research in database systems, pervasive/mobile computing, and networking. Prior to joining Pennsylvania State University, he was a principal member of the technical staff at Verizon/GTE Laboratories. He is particularly interested in developing data management techniques (including accessing, indexing, caching, aggregation, dissemination, and query processing) for supporting complex queries in a wide spectrum of networking and mobile environments such as peer-to-peer networks, mobile ad hoc networks, wireless sensor networks, and wireless broadcast systems. Meanwhile, he has worked on XML, security, information integration/retrieval, and object-oriented databases. His research has been supported by the US National Science Foundation (NSF) and industry grants. Most of his research results have been published in prestigious journals and conference proceedings in the fields of databases, mobile computing, and networking. He has served as a guest editor for several journal special issues on mobile database-related topics, including the *IEEE Transactions on Computers*, *IEEE Personal Communications Magazine*, *ACM MONET*, and *ACM WINET*. He was the founding program committee cochair for the International Conference on Mobile Data Management. He is a member of the IEEE and the ACM.