Multimedia Systems (2007) 13:19–33
DOI 10.1007/s00530-007-0078-9

REGULAR PAPER

Sampling cluster endurance for peer-to-peer based content distribution networks

Vasilios Darlagiannis · Andreas Mauthe · Ralf Steinmetz

Published online: 10 March 2007

Abstract Several types of Content Distribution Networks are being deployed over the Internet today, based on different architectures to meet their requirements (e.g., scalability, efficiency and resiliency). Peer-to-peer (P2P) based Content Distribution Networks are promising approaches that have several advantages. Structured P2P networks, for instance, take a proactive approach and provide efficient routing mechanisms. Nevertheless, their maintenance can increase considerably in highly dynamic P2P environments. In order to address this issue, a two-tier architecture called Omicron that combines a structured overlay network with a clustering mechanism is suggested in a hybrid scheme.

In this paper, we examine several sampling algorithms utilized in the aforementioned hybrid network that collect local information in order to apply a selective join procedure. Additionally, we apply the sampling algorithms on Chord in order to evaluate sampling as a general information gathering mechanism. The algorithms are based mostly on random walks inside the overlay networks. The aim of the selective join procedure is to provide a well balanced and stable overlay infrastructure that can easily overcome the unreliable behavior of the autonomous peers that constitute the network. The sampling algorithms are evaluated using simulation experiments as well as probabilistic analysis where several properties related to the graph structure are revealed.

The original paper with the same title has been published in the proceedings of MMCN'2006. This is an extended version including further results invited for submission at ACM/Springer MMSJ.

V. Darlagiannis (✉)
Swiss Federal Institute of Technology (EPFL),
1010 Lausanne, Switzerland
e-mail: Vasilios.Darlagiannis@epfl.ch

A. Mauthe
Computing Department, Lancaster University,
Lancaster, LA1 4YR, UK
e-mail: andreas@comp.lancs.ac.uk

R. Steinmetz
Multimedia Communications (KOM), Technische Universität
Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany
e-mail: Ralf.Steinmetz@KOM.tu-darmstadt.de

1 Introduction

Content Distribution Networks (CDNs) are used to deliver content to potentially very large user populations [1]. Within the Internet, different types of CDNs are being envisaged ranging from simple Web based applications to sophisticated multimedia entertainment systems, including interactive systems such as multiplayer games and virtual environments. Whereas first generation CDNs have mostly focused on Web content, the current, second generation systems also deal with media delivery.

A special, very successful type of CDNs are peer-to-peer (P2P) file sharing systems. In 2001 for instance, Napster was the fastest growing application in the Internet's history [2]. Since Napster, a number of unstructured P2P systems have been developed such as Gnutella [3], eDonkey [4], as well as structured approaches such as Chord [5], CAN [6], etc. Structured P2P networks aim in maintaining a topology based on explicit rules (e.g., a hypercube) while unstructured are more freely evolving network structures (e.g., power-law networks). What these systems share is the idea to have independent, collaborating nodes that organize and share information in a peer-to-peer fashion. Ideally for

large expandability and freedom in interactivity, there is no central instance that polices or governs the interaction between the peers as is the case in client-server interaction. The P2P paradigm basically states that P2P systems are self-organizing systems consisting of equal, autonomous entities where the interaction is governed by rules. However, in reality P2P systems are composed of peers with heterogeneous characteristics and user behavior [7].

This paradigm can be applied to a multitude of structures and systems. Apart from file sharing, P2P mechanisms are also proposed for media (mainly video) streaming [8–11]. Here, the focus is on the streaming of media from multiple senders to one receiver exploiting certain media properties (e.g., layered video coding). P2P structures are also being used for the transmission of media to multiple receivers, as in the case of application level multicast. A single tree approach is for instance taken in PeerCast [12] and SpreadIT [13]. In order to achieve better load balancing and improve resilience to node failures, multiple multicast trees are employed in the case of P2PCast [14] and SplitStream [15]. Other questions that are being addressed in the context of P2P based content distribution networks is the replication of files on a large set of peers. This has been labeled Quality of Availability (QoA) and defines a metric for the availability of certain content items within the system [16]. BitTorrent [17] is one of the most popular systems using replication on a wider scale. Though it uses a central instance (i.e., a web-service to redirect the client to the tracker) the exchange and organization of content is essentially P2P. FastReplica [18] is another replication system for large scale replication that uses a central instance, comparable to the control of surrogate servers in the case of Content Distribution Interworking (CDI).

The problem most P2P based CDNs encounter is the multitude of requirements placed on them. Hence, a number of solutions are very restricted in their approach concentrating on a (sub-)set of the critical requirements. However, this only provides a solution for specific cases and thus, they cannot be applied more widely. In order to build more generic CDN infrastructures based on P2P principles that maintain the advantages of P2P (such as flexibility and dynamicity), it is necessary to take certain requirements into account. Such a system has to be, for instance, able to cope with the inherent heterogeneity of peers since a common denominator approach would make it very inefficient. Further, it has to provide scalability, be incrementally expandable and dependable in its service. It should also balance the load of requests in a way that no hot-spots occur. The aim is to develop principles and methods that can be used to build P2P based content infrastructures that can be used in a multitude of environments and cases. Eventually, the goal is to create a generic infrastructure that can support all kinds of multimedia applications, including content production and delivery networks, interactive multimedia applications, multiplayer games, etc.

This paper elaborates on a particular issue in designing overlay networks, the network stability issue. Since peers are autonomous entities, they dynamically participate in the constructed network by joining and leaving. In fact, several empirical observations of the uptime distribution (c.f. [19,20]) indicate that the majority of peers do not stay connected for long time periods. Therefore, structured approaches utilizing proactive mechanisms in order to provide efficient routing mechanisms, require significant signaling to maintain the targeted topology and update the indexing data on the advertised content. The required information exchanged in this process can be further increased if the proactive design of the system replicates the content, too. Omicron (*Organized Maintenance, Indexing, Caching and Routing for Overlay Networks*) [21] addresses this issue with a two-tier architecture combining a structured approach with a clustering mechanism. Omicron constructs clusters of peers that (as a set) form reliable components to develop a stable structured overlay.

In order to do so, new peers perform a *selective join* mechanism, so that the resulting joint reliability of each cluster is above a minimum threshold. Generally defining, selective join is the procedure of sampling a number clusters gathering their properties and then selecting to join the cluster that meets better the predefined requirements. The joint reliability of a cluster is called *endurance* to reflect the differences from single peer reliability and network stability. The selective join mechanism is based on random sampling to select a subset of clusters in order to decide which one is the weakest from this subset. Over time, this has the effect of strengthen the weakest. The sampling technique is implemented by simply initiating random walks into the network and collecting the local properties as the messages are forwarded to their neighbors (messages increase in size on each hop). Several algorithms have been investigated in order to evaluate their performance on cluster coverage and the related properties.

Stable P2P networks provide the required infrastructure for different kinds of CDNs to operate efficiently. Omicron's design provides the additional aforementioned requirements, such as being scalable, incrementally expandable and dependable, providing evenly distributed workload properties, dealing adaptively with potential hot-spots, supporting heterogeneous populations, etc. However, sampling can be used as an effective mechanism for gathering several types of information in

P2P systems. In fact, sampling can be effectively applied in cases where probabilistic techniques can outperform their deterministic counterparts. Therefore, the evaluation of the proposed algorithms on other P2P networks can provide generalized results of greater interest. Chord [5] is a well-known P2P network with appealing scalability properties, which is used as a test-bed for our algorithms in addition to the Omicron-based experiments.

This paper is organized as follows: In Sect. 2, an overview of the Omicron network is provided focusing on the graph structure, the clustering mechanism and the resulting architecture. The network management mechanism dealing with peers joining the network is discussed in Sect. 3. The investigated sampling algorithms are provided in Sect. 4, and the related simulation results are given in Sect. 5 together with probabilistic analysis of the most critical aspects. The related work is provided in Sect. 6. Finally, Sect. 7 summarizes the paper and gives an outlook for the future.

2 Omicron

Omicron is a P2P overlay network aiming to address issues of heterogeneous, large-scale and dynamic P2P environments. Its hybrid, two-tier, DHT-based approach makes it highly adaptable to a large range of applications. Omicron deals with a number of conflicting requirements, such as scalability, efficiency, robustness, heterogeneity and load balance. Issues to consider in this context are:

Topology. The rational in Omicron's approach is to reduce the high maintenance cost by having a small and fixed node degree, thus, requiring small and fixed size routing tables (at least for the majority of peers), while still performing lookup operations at low costs. For this reason, the usage of appropriate graph structures (such as de Bruijn graphs [22], which are further discussed in Sect. 2.1) is suggested. However, while the small fixed node degree reduces the operational cost, it causes robustness problems.

Clustering mechanism. To address the robustness issue, clusters of peers are formed with certain requirements on their endurance. The clustering mechanism is described in deeper detail in Sect. 2.2.

Roles. A unique feature of Omicron is the integrated specialization mechanism that assigns particular roles to peers based on their physical capabilities and user behavior. The specialization mechanism provides the means to deal with peer heterogeneity. This scheme fits the contribution of each node to its resource capabilities and aims at the maximization of the cluster efficiency by providing appropriate incentives to peers to take a certain role. As it can be observed from Omicron's name, four different core roles have been identified: *Maintainers* (*M*), *Indexers* (*I*), *Cachers* (*C*) *and Routers* (*R*). Maintainers are responsible to maintain the overlay network topology, while Indexers handle the relevant indexing structures. Routers forward the queries towards their logical destination, and Cachers reduce the overall routing workload by providing replies to popular queries. Roles are additively assigned, meaning that peers do not remove their older roles as they get new ones.

Identification scheme. Since Omicron is based on clusters, there is a need to identify both the clusters and the peers. Therefore, a *dual* identification scheme is proposed to satisfy the identification requirements. Peers are distinguished by a (GUID) that is created using secure hash functions. Such peer GUIDs have *constant* length. Clusters are distinguished by *dynamically* modified GUIDs that follow a de Bruijn-like value assignment. The peculiarity of this scheme is that the length of the identifiers is adapted to the size of the graph. Moreover, the length of de Bruijn identifiers may differ by maximum one for neighbor nodes in order to fulfill the requirement of incremental expandability and of evenly distributed workload. The benefits of this dual identification scheme are multi-fold. Nodes can be uniquely identified and their actions may be traced when this is desirable. Thus, peers are responsible for their actions. In addition, peers cannot "force" responsibilities for indexing certain items since the mapping is not depending on the peer GUID but on the cluster GUID. Further, neighbor selection is not strictly defined. Thereby, peers can select their neighbors from neighbor clusters. This selection may be based on network proximity, trust or other specific metrics.

2.1 de Bruijn Digraphs

Directed graphs (digraphs) have been extensively used in interconnection networks for parallel and distributed systems design (cf. [23,24]). Digraphs received special attention from the research community aiming to solve the problem of the so-called (k, D) digraph problem [25], where the goal is to maximize the number of vertices (order) N in a digraph of maximum out-degree k and diameter D. Some general bounds relating the order, the degree and the diameter of a graph are provided by the well-known Moore bound [26]. Assume a graph with node degree k and diameter D; then the maximum number of nodes (*graph order*) that may populate this graph is given by Equation 1:

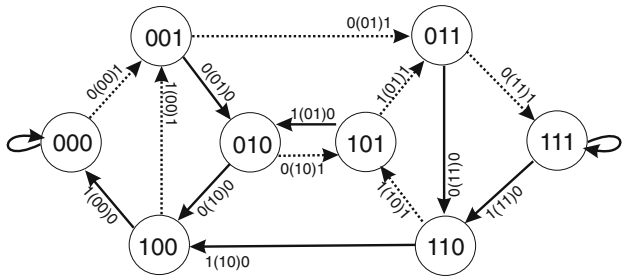

Fig. 1 Directed de Bruijn(2, 3) graph

$$N \leq 1 + k + k^2 + \cdots + k^D = \frac{k^{D+1} - 1}{k - 1}. \quad (1)$$

Interestingly, the Moore bound is not achievable for any non-trivial graph [26]. Nevertheless, in the context of P2P networks, it is more useful to reformulate Equation 1 in a way that provides a lower bound for the graph *diameter* (D_M), given the node degree and the graph order [27]:

$$D_M = \lceil \log_k(N(k-1)+1) \rceil - 1 \leq D. \quad (2)$$

The *average distance* (μ_D) among the nodes of a graph may also be bounded by the following inequality [28] (which is approximated by Loguinov et al. [29]):

$$D_M - \frac{k(k^{D_M} - 1)}{N(k-1)^2} + \frac{D_M}{N(k-1)} \approx D_M - \frac{1}{k-1} \leq \mu_D. \quad (3)$$

An interesting class of digraphs is the so-called lexicographic digraph class [30], which includes the de Bruijn and Kautz digraphs.[1] de Bruijn digraphs have asymptotically optimal graph diameter and average node distance [29]. Thereby, they are employed in the design of our work. de Bruijn graphs have been suggested to model the topology of several P2P systems, however, we exploited them in an innovative way. Considerable examples of P2P systems that use de Bruijn graphs are Koorde [31], D2B [32] and Optimal Diameter Routing Infrastructure (ODRI) [29].

Figure 1 shows a directed de Bruijn(2, 3) graph denoting a graph with a maximum out-degree of 2, where the diameter length is 3 and the graph order is 8. For graphs with fixed out-degree of 2, the maximum number of nodes[2] is always limited by 2^D. The graph contains 2^{D+1} directed edges in this case. Each node is represented by string of length D ($D = 3$ in this example). Every character of the string can take k different values (2 in this example). In the general case, each node is represented by a string such as $u_1u_2...u_D$. The connections between the nodes follow a simple left shift operation from node $u_1(u_2...u_D)$ to node $(u_2...u_D)u_x$, where u_x can take one of the possible values of the characters $(0, k-1)$. The shifted-in character determines the selected neighbor to follow in the routing procedure. The solid lines in the figure denote links where the '0' character is shifted in, while the dotted lines denote links where the '1' character is shifted in.

[1] de Bruijn graphs are less dense than Kautz graphs but they are more flexible since they do not have any limitations on the sequence of the represented symbols in every node.

[2] The Moore bound determines always maximum upper bounds on the size of the graphs that are not reachable for non-trivial cases.

2.2 Clustering

Clusters have been introduced into the design of P2P systems in a variety of approaches. JXTA defines the concept of PeerGroups [33] to provide service compatibility and to decompose the large number of peers into more manageable groups. Further, SHARK [34] cluster peers based on the common interests of users. Also, Considine [35] proposes multiple cluster-based overlays for Chord. The cluster construction in the latter proposal is based on network proximity metrics aiming to reduce the end-to-end latency. Furthermore, even hierarchical approaches like eDonkey and KaZaA might be considered as clustering approaches to a certain extent, where normal peers are clustered around the super-peers. The purpose of this "clustering" is to transform the costly all-to-all communication pattern into a more efficient scheme. However, by doing so it introduces additional load-balancing concerns. In fact, this is a more general issue that appears in every acyclic hierarchical organization (i.e., tree-like organization). Thus, the cluster organization must be restricted to non acyclic structures in order to provide even distribution of responsibilities.

A desirable property of each overlay network is acquiring a topology that remains as *stable* as possible over time and minimizes the related required communication cost to maintain the targeted structure. However, in highly dynamic P2P systems that consist of *unreliable* peers, perfect stability cannot be attained. This is the most crucial motivating factor for introducing the concept of clusters in the architectural design of the Omicron overlay network. Clusters can be considered as an essential abstraction, which can be used to absorb the high peer attrition rate and accomplish high network stability. They can be considered as an equivalent mechanism to the suspensions used in vehicles to absorb shocks from the terrain. In order to make more clear the involved concepts, it is required to define *peer reliability*, *network stability* and *cluster endurance*. We define network stability as follows:

Definition 2.1 Network stability $S_N(t)$ is the probability that the topology of the network remains unmodified for some time t.

A definition for peer reliability is given below.

Definition 2.2 Peer reliability $R_P(t)$ is the probability that the peer remains connected for some time t.

Assuming that the lifespan of a peer is modeled with the random variable X, then the reliability of the peer is given by:

$$R_P(t) = Pr\{X > t\} = 1 - F(t). \tag{4}$$

where $Pr\{\cdot\}$ denotes the probability statement and F(t) is the Cumulative Distribution Function (CDF) of peer's lifetime over the set of peers active in the system at any given moment. This distribution is different from the one used to describe the peers' lifetime as they arrive in the system This is due to the fact that longer-lived peers spend more time in the system and therefore make up a larger fraction of (currently) active peers.

On the other hand, a cluster is a virtual entity composed by several peers. We define the endurance of a cluster as follows:

Definition 2.3 Cluster endurance $E_C(t)$ is the probability that at least one peer of the cluster will remain connected for some time t.

The endurance of clusters is calculated by the following equation.

$$E_C(t) = 1 - \prod_{i=1}^{K}(1 - F_i(t)), \tag{5}$$

where K is the size of the cluster and $F_i(t)$ is the CDF of the ith peer in the cluster.

Clustering algorithms aim mainly at *"partitioning items into dissimilar groups of similar items"*. They require the definition of a *metric* to estimate the similarity of the items in order to perform the partitioning procedure. Since an overlay network is a virtual network, there is a lot of freedom in defining the optimal partitioning metric. In the context of P2P overlay networks, the similarity of the peers forming an individual cluster is that all the members are responsible for the same part of the *address space*, which does not necessarily define a meaningful metric for assigning peers to clusters. Therefore, the proposed clustering algorithm requires criteria other than the usual similarity metrics. The key factors that motivate the construction/deconstruction of clusters are the following:

- Clusters should fulfill the endurance requirements.
- Clusters should have the smallest possible size in order to reduce the intra-cluster communication complexity.
- Clusters should be divided when it is possible to create d other endurable clusters, where d is the degree of the employed de Bruijn graph. Selective division should be applied to maximize the endurance of each new cluster.
- Clusters should be merged when their estimated endurance is lower than a predefined endurance threshold.

Moreover, a hysteresis-based mechanism is required to avoid oscillations in splitting and merging clusters. In order to describe the membership of peers in the clusters, the ClusterMap concept has been used. A *ClusterMap* includes the peers participating in a cluster. ClusterMaps may be realized as tables collecting entries for each member peer. Every entry may hold information about peers' GUID, their role in the system, the observed reliability and other useful information that could be used by every peer of a cluster to effectively construct its local routing table. In fact, ClusterMaps are supersets of Routing Tables, including the potential peers of clusters that may become neighbors of a particular peer. ClusterMaps are periodically disseminated to neighbor clusters as well as the cluster itself.

2.3 Two-tier network architecture

The suggested *two-tier* network architecture is a major step towards accomplishing the fulfillment of the targeted requirements. It enables the effective usage of de Bruijn graphs by successfully addressing their shortcomings, such as inflexible network expandability and low network resilience for low node degree. In fact, the successful "marriage" of two different topology design techniques (in a combination of a *tightly structured macro level* and a *loosely structured micro level*) provides a hybrid architecture with several advantages.

1. **Tightly structured macro level.** Adopting the topological characteristics of de Bruijn graphs, the macro level is *highly symmetrical* enabling *simple routing* mechanisms. Composed of endurable components, it results in a relatively *stable* topology with *small diameter* and *fixed node degree*.
2. **Loosely structured micro level.** On the other hand, the micro level provides the desirable characteristics to the macro level by following a more loosely structured topology with a great degree of *freedom*

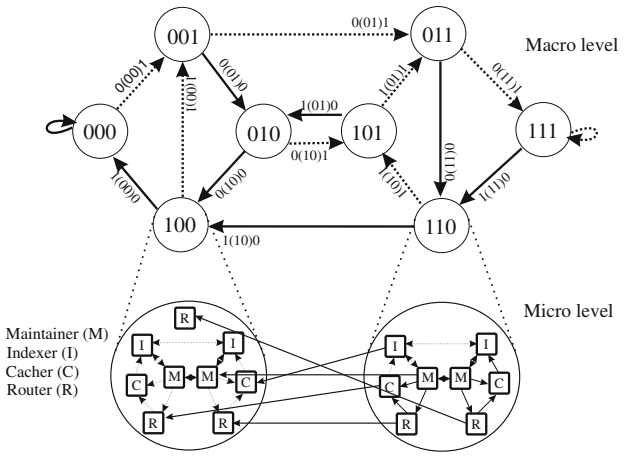

Fig. 2 Omicron overlay network

in the neighbor selection. This freedom may be invested on regulating and achieving a *finer load balance*, offering an effective mechanism to handle potential *hot spots* in the network traffic. Moreover, *locality-aware* neighbor selection may be used to maximize the matching of the virtual overlay network to the underlying physical network. Finally, *redundancy* may be developed in this micro level supplying seamlessly *fault-tolerance* to the macro level.

An example of the hybrid topology is illustrated in Fig. 2. The structured macro level is a de Bruijn(2, 3) digraph. Two nodes (representing peer clusters) are "magnified" to expose the micro level connectivity pattern between them. Two different connection types are shown: inter-cluster connections and intra-cluster connections. Further information (e.g., on the relevant routing mechanism) can be found at [7].

3 Overlay network management

In the resulting two-tier architecture, two entities are used to construct the network topology: individual peers and clusters of peers. Thereby, a set of efficient procedures need to be defined to handle the dynamic participation of the peers in the system and the resulting consequences in both the endurance and the maintenance cost of clusters. Three crucial requirements are driving the developed solutions: *dependability*, *load-balance* and *efficiency*.

When new peers request to join an Omicron-based P2P system, Maintainers perform a number of operations in order to place the new peers in the network. The purpose of their actions is to achieve a well balanced topology where clusters have sufficient endurance and the total workload is minimized and well distributed.

Obviously, the optimal selection can be made when the endurance of every cluster of the network is globally known (or at a particular central entity). However, such a solution raises scalability issues as the size of the network increases considerably. Therefore, an alternative approach has been investigated where Maintainers perform a random walk collecting the endurance of each cluster in the path in order to decide which one is the best selection to direct the new peer to. In addition to the random walk based solution, we have considered a publish/subscribe mechanism where low endurance clusters become publishers. In this case, all the clusters of the network should subscribe to these events, which makes the solution unscalable. An advanced source to destination assignment will increase significantly the complexity of the solution in contrast to the elegant probabilistic approach of random walks.

A variety of bootstrap phases may be assumed, providing an initial online peer P_Y that triggers the mechanism to accept the newly joining peer P_X. Without loss of generality it can be assumed that P_Y has been assigned the Maintainer role (otherwise the request has to be simply redirected to another peer $\acute{P_Y}$ of the same cluster that has been assigned the Maintainer role).

Upon the reception of the joining request P_Y triggers a *sampling* procedure using an inter-cluster *random walk*. It contacts a Maintainer P_Z of a randomly selected neighbor cluster, which is recursively repeating this step making a random walk of length $w = \alpha \cdot log(C_S)$, where C_S is the number of clusters in the system and α is a weight. It should be noted that w is asymptotically equal to the diameter of the inter-cluster overlay network. The goal of the procedure is to equally distribute the new peers in the deployed clusters considering the internal state of each cluster, i.e., its endurance and its size. By performing a random walk of length at least equal to the diameter of the network, every cluster has a non-zero probability of being included in the sampling procedure of each join request. This probability is related to the cluster location in the overlay network. Moreover, having a logarithmic number of samples provides a fairly good approximation of collecting the state of all clusters, which otherwise, it would have been very costly in terms of communication traffic to obtain accurately. Thus, the selected approach can provide a well-balanced outcome with a low cost. Finally, as it will become more clear in Sect. 5.3, such sampling based on random walks of logarithmic length has statistical properties similar to independent sampling.

After performing the sampling procedure with the random walk, a suitable cluster is selected for joining. In

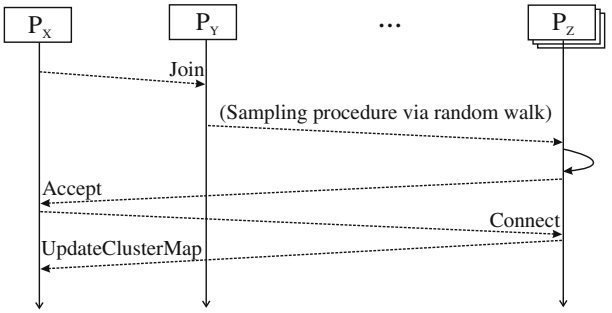

Fig. 3 Join sequence diagram

this phase, a newly joined peer is considered unreliable, and it is merely assigned the Router (and optionally the Cacher) role. Thus, the selected cluster is the one with the least number of unreliable peers so that the load for the Maintainers of the clusters is fairly equal. The Maintainer of the last visited cluster included in the random walk indicates to the newly joined peer the selected cluster of which it should become a member of (i.e., via an `Accept` message). Afterwards, P_X asks the provided Maintainer of the target cluster to connect and become a cluster member. As a reply, the Maintainer provides updated ClusterMap structures of the cluster itself and the neighbor clusters so that the P_X can correctly build its routing table.

The whole process is illustrated in Fig. 3 by a sequence diagram. Peer P_Z represents the Maintainers that participate in the sampling random walk. The selected Maintainer receives the *Connect* message and replies by providing the necessary ClusterMap structures. A further issue related to the network management is the way the structured macro level expands or shrinks in order to fit to the network size. For this purpose, a decentralized algorithm to split and merge the clusters is described in [21]. Moreover, a similar random walk mechanism may be applied when peers are becoming reliable enough to be assigned more critical roles (i.e., Indexers and Maintainers). In this case, a migration phase takes place ensuring the better distribution of the reliable peers among the clusters.

4 Investigated sampling algorithms

In this section, we describe four different algorithms to accomplish effective cluster coverage at low cost. Three of them are probabilistic and one is deterministic. All algorithms start randomly at any peer, assuming that new peers randomly select their first contact to send the requests to. Even if the employed bootstrap phase does not comply with this assumption, it can be easily achieved by performing an additional random walk before the sampling phase begins. Sampling is performed by initiating the submission of a message to a randomly (or deterministically) chosen neighbor cluster. The GUIDs and the relevant endurance values of the visited clusters are collected in the body of the message itself. This random walk terminates following the rules defined by the particular algorithms.

4.1 Probabilistic algorithms

The probabilistic algorithms differ both in length and neighbor selection policy. Their description is provided in the following list.

1. **Random destination**. This algorithm starts from any random peer, which randomly selects the final destination. This has the advantage of simplicity since it does not differ from a typical query routing procedure. However, the number of covered clusters is equal to the average query length. Therefore, the average random walk length is given by Equation 3, which is shorter than the network diameter. The achieved cluster coverage is very similar to the assigned routing workload assuming uniform query distribution [7].
2. **Short random walk**. This algorithm starts from any random peer by randomly selecting only the next peer to follow among the neighbors found in the routing table. The procedure is recursively applied until a random walk of length equal to the diameter D of the network is reached. This algorithm has the advantages of (i) equal length random walks and (ii) better coverage distribution than the previous algorithm since seldom reached clusters are more likely to be visited. However, its implementation is more complex. It requires a non-oblivious routing mechanism to avoid cycles in the random walk. Clusters should be visited only once for efficiency, however, this rule can be ignored for particular topologies where it will not allow visiting enough nodes.
3. **Long random walk**. This algorithm is very similar to the previous one. The only difference is that the required length of the random walk must be twice the length of the diameter ($2D$). It is expected that the longer random walk combined with the cycle avoidance restriction will provide a much better cluster coverage. The disadvantage of this algorithm is that it costs twice as much as the short random walk.

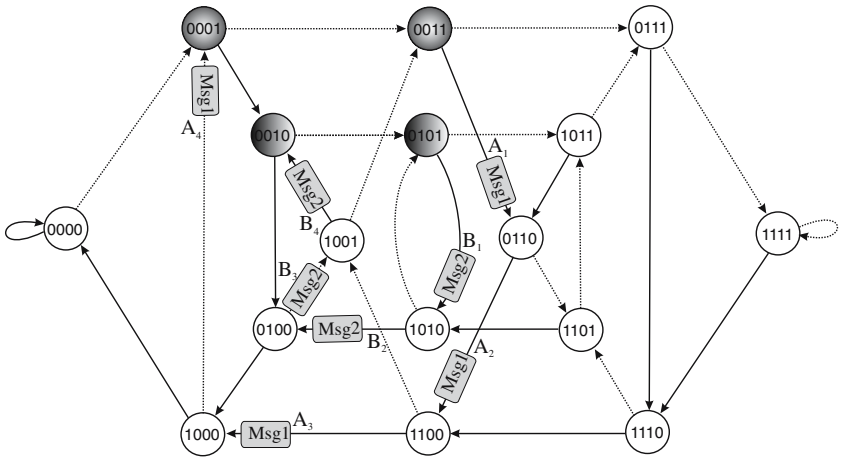

Fig. 4 Deterministic *R*-shift algorithm

4.2 Deterministic *R*-shift algorithm

The aforementioned algorithms perform random walks in order to sample the endurance of the clusters. In this section, a deterministic algorithm is investigated in order to evaluate such an alternative. It is assumed that the deterministic walk begins randomly at any peer (similarly to the random alternatives).

There are certain restrictions and guidelines in designing an effective deterministic walk appropriate to effectively sample the endurance of clusters.

1. The length of each walk must be as close as possible to its maximum value (i.e., the diameter of the network).
2. The length of each walk should not differ considerably (independently of the position of the initial cluster).
3. The cluster coverage should be as wide and as evenly distributed as possible.

There are several algorithms that can fulfill the aforementioned requirements. We have designed one that is as simple as possible. It is called *"R-Shift"* algorithm. Basically, each peer deterministically select the final destination by applying a *right-shift* operation at the GUID of its cluster (note that the conventional routing in Omicron utilizes *left-shift* operations). The new symbol at the left end of the GUID must be different than the symbol at the right end before the right-shift operation. Formally, this operation can be expressed as follows.

$$r\,\text{shift}(u_1u_2...u_D) = \overline{u_D}u_1u_2...u_{D-1}, \tag{6}$$

where the $\overline{u_D}$ is an operation that provides a different symbol from the available alphabet (deterministically). For example, for binary de Bruijn graphs, it holds that $\overline{0} = 1$ and $\overline{1} = 0$. It is guaranteed that using the R-Shift algorithm all the clusters will be included in the sampling since Equation 6 provides a direct and unique mapping of the input cluster to the output cluster.

This algorithm is graphically illustrated in Fig. 4, which displays a de Bruijn(2, 4) digraph. Two different deterministic walks are shown. Assume that the two walks start at nodes (0011) and (0101), respectively. Thereby, the *R*-shift algorithm produces the sequence (0011) $\rightsquigarrow$ (0110) $\rightsquigarrow$ (1100) $\rightsquigarrow$ (1000) $\rightsquigarrow$ (0001) for the first case, which is traversed by `Msg1`. Similarly, the sequence (0101) $\rightsquigarrow$ (1010) $\rightsquigarrow$ (0100) $\rightsquigarrow$ (1001) $\rightsquigarrow$ (0010) is traversed by `Msg2`. However, in certain cases the length of the path is smaller than the diameter, e.g., (1101) $\rightsquigarrow$ (1011) $\rightsquigarrow$ (0110).

The described algorithms have different inter-cluster communication cost that determines the sampling walk length. Table 1 summarizes this cost.

Table 1 Sampling algorithms routing cost

Sampling algorithm	Expected walk length
Random destination	$D_{DB} - \frac{1}{k-1}$
Short random walk	D_{DB}
Long random walk	$2 \cdot D_{DB}$
R-shift	$\approx D_{DB}$

5 Evaluation and analysis

The evaluation of the sampling algorithms has been performed by simulation experiments using the general purpose discrete event simulator for P2P overlay networks described in [36]. In most of the experiments, the constructed overlay network forms an Omicron network, however, in several cases we additionally observe the sampling properties in Chord [5] overlay networks.

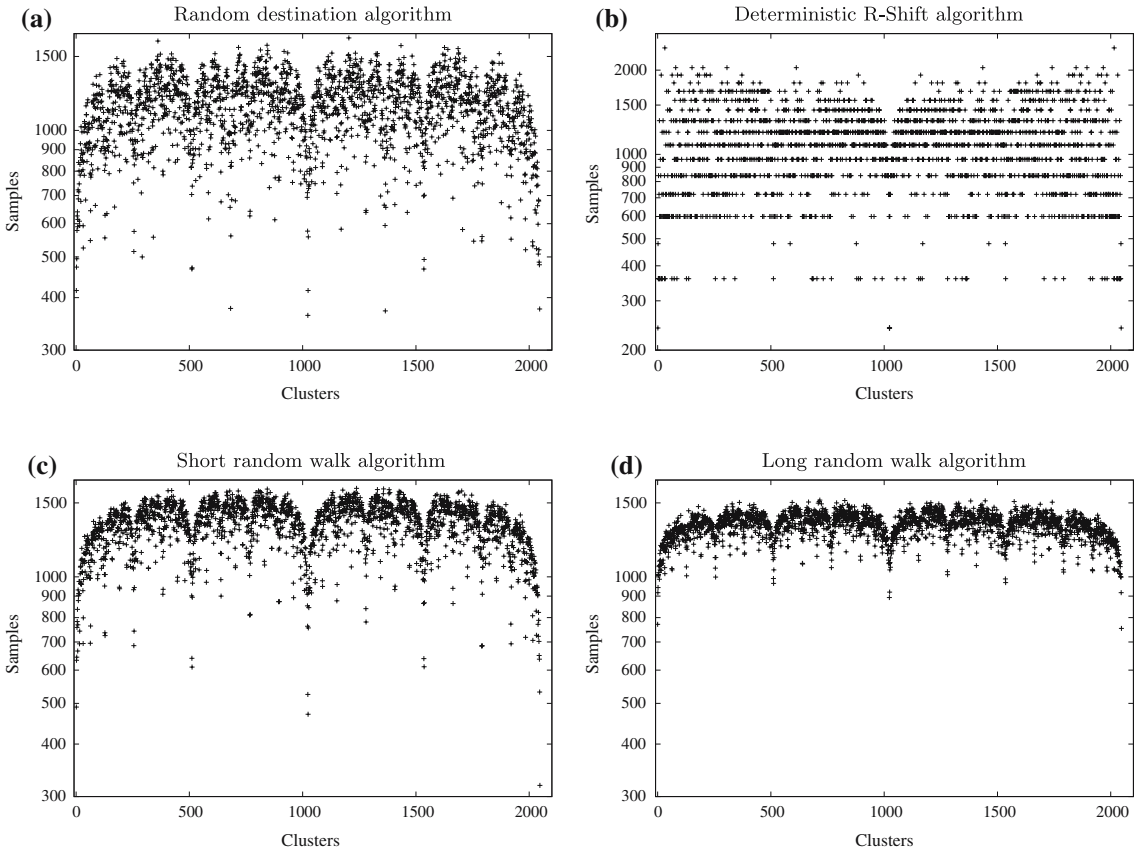

Fig. 5 Sampling distribution using random walks

Chord and Omicron differ in the node degree and the symmetry properties. Both networks have been stabilized before the sampling procedures. Moreover, a probabilistic analysis is engaged in some aspects to verify to observed simulation results. During the simulated experiments, message delivery considers the physical distance between the nodes enhanced by a statistical queue model. However, connection bandwidth limitations have been ignored, assuming that the involved signaling protocol is light enough and does not generate large traffic.

5.1 Cluster Coverage

The most critical aspect we need to evaluate is the ability of the four sampling techniques to visit evenly each cluster of the network. This case is more interesting for the de Bruijn digraph based network that lacks node symmetry. Figure 5 provides the results for a specific Omicron configuration where the structured de Bruijn network is composed of 2,048 clusters and the inter-cluster degree is $k = 2$. Figure 5a describes the coverage distribution (the number of times each cluster is sampled) for the random destination algorithm. Similar results are provided in Fig. 5b–d for the *R*-shift algorithm, the short random walk algorithm and the long random walk algorithm, respectively. Furthermore, it can be observed that the results are symmetrical among the left half and the right half of the figures (patterns can be observed). This is the result of the symmetry properties of the de Bruijn digraphs combined with the even, deterministic distribution of the sampling initiation among the clusters.

As expected, the long random walk algorithm provides the most evenly distributed sampling where the majority (in most of the experiments over 90% of the population) of samples differ less than 10% from the mean value. The reason for this lies on the fact that more clusters are sampled at every join, which is combined with the cycle-removal mechanism (non-oblivious routing mechanism). Therefore, clusters that are seldom sampled by other algorithms have a higher probability of sampling by this algorithm.

Aiming at providing additional results on the ability of each algorithm to evenly sample the network clusters, further experiments have been performed. In

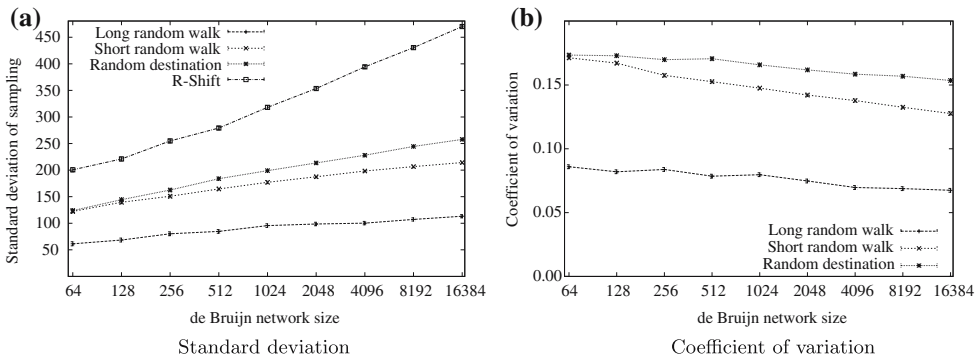

Fig. 6 Cluster sampling in de Bruijn networks

these experiments, the size of the structured macro level (de Bruijn network) is modified between 64 and 16,384 clusters. For each experiment, a number of join requests is generated that is related to the size of the network and the utilized algorithm. The target is to generate approximately equal workload for all of the algorithms.

The quantity of interest in these experiments is the evaluation of the *standard deviation* of the cluster sampling distribution for each algorithm. Figure 6a summarizes the results of the experiments. It should be noted that the x-axis scales logarithmically in order to provide a more comprehensive view. As it can be observed, the long random walk algorithm achieves the smallest standard deviation for the complete range of the evaluated network sizes. Moreover, the short walk algorithm achieves better performance compared to the random destination algorithm as the network grows. The standard deviation of the R-shift algorithm grows considerably more compared to the three probabilistic alternatives.

However, the standard deviation provides an absolute value for the effect. In many cases, it is more important to observe a relative metric that relates the standard deviation with the mean value. Such a metric is the *coefficient of variation*, which is defined as $\mathrm{CV} = \sigma/\mu$, where σ is the standard deviation and μ is the mean value. It should be noted that the mean value of the deterministic algorithm differed from the provided mean values set. Therefore, it is not included in the provided results. Figure 6b summarizes the results on the coefficient of variation. As it can be observed, the long random walk algorithm accomplishes always a coefficient of variation less than 10%. Also, it is interesting to notice the decreasing rate of CV for the short random walk algorithm. It can be stated that for very large network size (as network size approaches infinity), the performance of the short random walk algorithm approaches asymptotically the performance of the long random walk algorithm.

5.2 Cluster sampling inter-arrival distribution

The aggregated behavior of the cluster sampling algorithms can be observed with the experiments performed in the previous sections. However, it is interesting to evaluate how often each particular cluster is revisited in subsequent random walks.

Self-connected clusters (i.e., with GUID 11...1 or 00...0) are the least frequently involved clusters in the routing procedure [7], which can also be observed in Fig. 5. Further, by examining the details of the collected results, it has been noticed that clusters with GUID lacking of "patterns" in their digit sequence, e.g., (011001001) or (0110010011) or (01100100110), are among the most frequently visited clusters for each sampling algorithm.

Let us define *each random walk experiment* as a *"round"*. The *event* of interest E_C is *"how many random walks* (*rounds*) *are necessary until a particular cluster C is sampled"*. Such a quantity can provide vital information on whether the sampling algorithm is adequate for its need. Therefore, further experiments have been performed aiming to evaluate E_C. Collecting the experiment results for these clusters, the diagrams of Fig. 7 have been drawn to show the probability that the particular cluster will be sampled after a certain number of sampling walks. The different location in the de Bruijn graph of nodes (0000000000) and (01100100110) (provided in Fig. 7a,b, respectively) results in different rates, following the same shape though. In order to generate these measurements, the long random walk algorithm has been employed.

It is interesting to observe that the cluster sampling inter-arrival distribution can be closely approximated by an *exponential* distribution of the form $f(x) = \lambda x^{-\lambda x}$, $x \geq 0$. The reason for such behavior can be explained as follows. Let us call V_C the event of interest, which is visiting a particular cluster C during a random walk. V_C is a *Bernoulli* random variable:

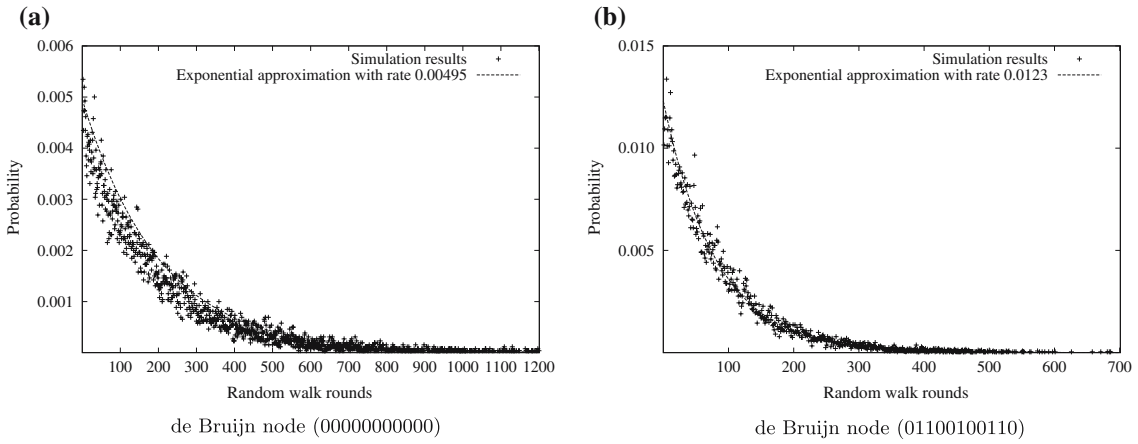

Fig. 7 Cluster sampling inter-arrival distribution on Omicron

$$g(x) = \begin{cases} g(0) = Pr\{V_C = 0\} = 1 - p, \\ g(1) = Pr\{V_C = 1\} = p, \end{cases} \quad (7)$$

where p is the probability of success that depends on the cluster position in the digraph. Each random walk is an independent event. If we let X be the number of the performed events until a success occurs, then X is said to be a *geometric* random variable with parameter p. Its PMF is given by:

$$h(n) = Pr\{X = n\} = (1 - p)^{n-1}p, \quad n = 1, 2, \ldots \quad (8)$$

The geometric distribution is the discrete equivalent of the exponential distribution. Therefore, the cluster sampling inter-arrival distribution can be approximated well with the exponential distribution.

As it can be seen from the approximated rates of Fig. 7, seldom visited clusters have a lower rate than frequently visited clusters. Also, as the size of the network gets larger, the approximated rates are getting smaller, which is expected since more clusters are available for sampling. However, the peer join rate is getting higher, providing the necessary lower bound for the sampling rate. In theory, there is a probability that a cluster might be under-sampled for a certain period, however, this is a worst-case scenario that does not occur often in practice.

The basic reason of the different inter-arrival rates in the aforementioned nodes is the asymmetry of the de Bruijn networks. On the contrast, Chord has a node-symmetric graph (the graph looks the same independently of the observation node). Therefore, we would expect to observe similar sampling inter-arrival rate for each Chord node. We have constructed such simulation experiments performing random walks in a Chord like-network consisting of 2,048 nodes. The network has been stabilized before the sampling procedure. At each node, a finger[3] is randomly selected and followed unless it has been sampled already during this walk (a new selection is made in this case). The results are shown in Fig. 8 where two different nodes have been randomly selected. It can be observed that similar to the Omicron case, inter-arrival distribution while sampling in Chord networks has similar exponential distribution properties. Moreover, all of the nodes have the same exponential factor as a result of the higher symmetry found in the Chord networks.

5.3 Cover-time

In this section, we examine the time it takes to sample each cluster at least once, the so-called *cover-time*. We measure the cover-time in number of sampling random walks required, as each one is triggered, i.e., when a new peer joins the network or a peer migration is necessary.

In order to model the aforementioned problem and provide a probabilistic evaluation, a useful result found in the related literature is utilized. Gkantsidis et al. [37] proves that as long as the random walk length is $O(\lg N)$ and the graph is an expander, the samples taken from the consecutive steps can achieve statistical properties similar to independent sampling. Since de Bruijn graphs are well-known expander graphs [38], the aforementioned results holds on them too.

Moreover, Feller [39, p. 225] provides a probabilistic analysis of *Coupon Collection Problem*. The Coupon Collection Problem is the following: Suppose that there are N distinct types of coupons, uniformly distributed.

[3] A finger is shortcut connection as it is defined for Chord networks [5].

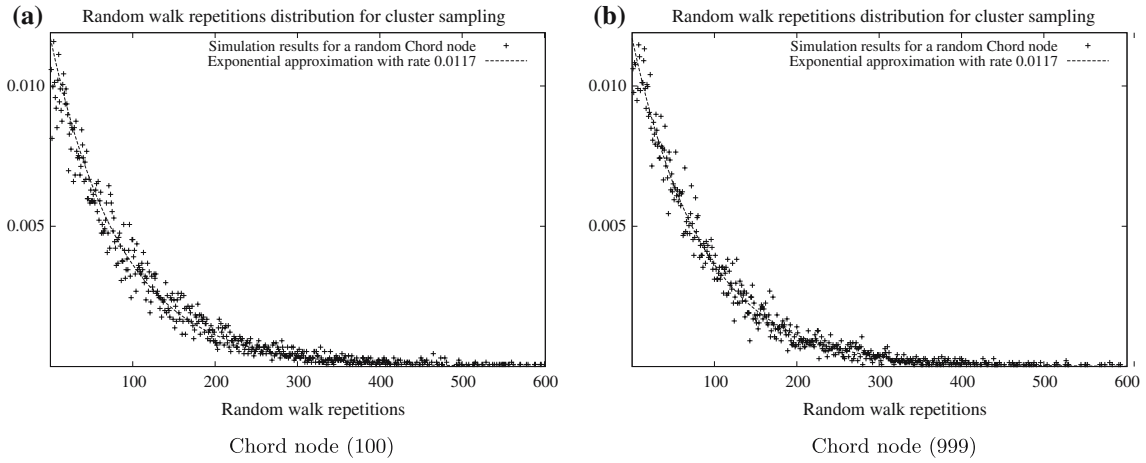

Fig. 8 Cluster sampling inter-arrival distribution on a Chord network

At each experiment step, we draw a coupon. Let S_r be the time by which r distinct coupon types have been encountered. It holds that[4]:

$$E[S_r] = N \cdot \lg\left(\frac{N}{N-r}\right) + O(N). \tag{9}$$

Equation 9 provides the expected number of single samples to cover each different sample type based on independent sampling. In our problem, we need to evaluate the number of necessary random walks of length k that suffice to sample each cluster of the network. Therefore, if we group k consecutive samples as a random walk, Equation 10 provides an approximation of the expected number of random walks (we ignore terms of order $O(N)$):

$$E[S_r] = \frac{N}{k} \cdot \lg\left(\frac{N}{N-r}\right). \tag{10}$$

We need to empirically verify the suitability of Equation 10 in predicting the expected number of random walks. Therefore, we have performed simulation experiments both on Omicron-based as well as on Chord-based overlay networks. In both overlays, we have considered networks of 2,048 nodes to be sampled. The number of samples collected in each random walk are $k = s+1$, where s is the random walk size, since the initiator of the random walk is also encountered in the sampling process.

As it can be observed from Fig. 9a,b where random walks of length $\lg N$ and $2 \cdot \lg N$ are, respectively, evaluated, the expected probabilistic cost of independent sampling fits accurately with the random walk based mechanism.[5] However, in the case where the length of the random walk is getting significantly less than the diameter of the graphs, the difference between the expected probabilistic cover time (based on independent sampling) and the simulation experiments based on random walks is becoming apparent. The results are shown in Fig. 9c,d demonstrating the random walks of length 2 and 3, respectively.[6] Nevertheless, despite their differences in the degree and symmetry, both Chord and Omicron perform similarly well with respect to the expected cover time (almost indistinguishable). The great majority of nodes is expected to be often sampled, which leads to high network stability.

Finally, some further experiments were contacted examining the performance of randomly selecting the next hop of a random walk in comparison with an alternative adaptive mechanism where less frequently selected neighbors have higher probability to be followed in future walks (history log is maintained). Surprisingly, no improvement was noticed both for Chord and Omicron with respect to cover-time performance.

[4] In this problem, a drawing event is successful if it results in sampling a new node. S_r is the number of sampling iterations until r successes. Let $X_k = S_{k+1} - S_k$. Then, $X_k - 1$ is the number of unsuccessful drawings between the kth and $(k+1)$st success. During these drawings, the population contains $N - k$ nodes that have not yet been sampled. Therefore, $X_k - 1$ is the number of failures preceding the first success in Bernoulli trials with $p = (N-k)/N$, alas, it holds that $E[X_k] = 1 + (1-p)/p = N/(N-k)$. Since $S_r = 1 + X_1 + \cdots + X_r$, the expectation is $E[S_r] = N\{\frac{1}{N} + \frac{1}{N-1} + \cdots \frac{1}{N-r+1}\}$. A useful approximation is given by $E[S_r] = N \cdot \lg \frac{N+1/2}{N-r+1/2} = N \cdot \lg \frac{N}{N-r} + O(N)$ [39, p. 225].

[5] Note that since the confidence interval bounds are very small, we have omitted them to increase the readability of the images.

[6] Consider that the network diameter in this case is $\lg N = 11$.

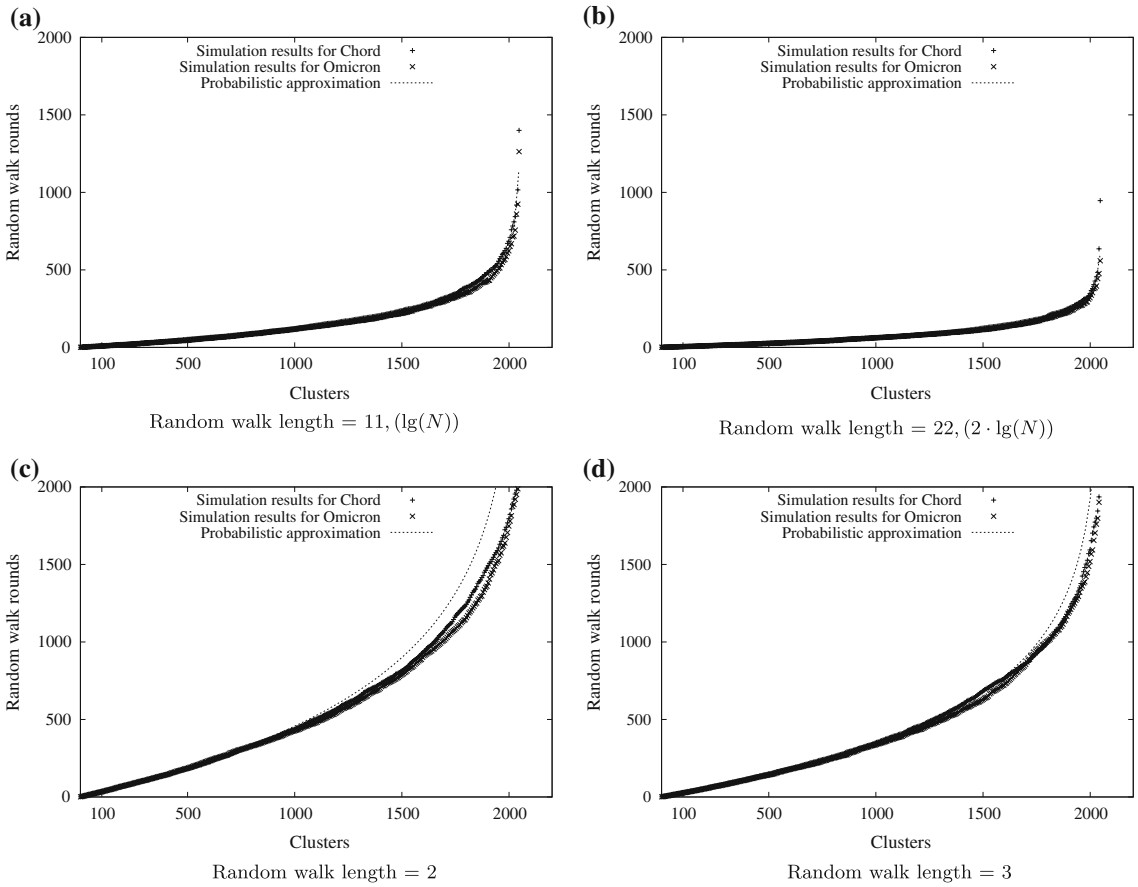

Fig. 9 Cover-time

6 Related work

Churn, the continuous process of nodes arrival and departure, is one of the basic issues that should be handled by DHTs, since it determines the network maintenance cost. A number of measurements reports (i.e., [19,20,40–42]) have investigated deployed P2P systems, most of them being file sharing applications. Despite the fact that the results reveal slightly different uptime node distributions, they all agree that the majority of the peers stay online for a relatively short time resulting in unstable networks. Therefore, proactive approaches such as DHTs require high maintenance cost to provide efficient routing of the lookup queries. This issue has been investigated by several other researchers, too. Here we summarize some of the most important related work.

Lam et al. [43] considered this problem and addressed it with two protocol extensions for hypercube-based approaches. The aim is to maintain *K-consistency* for the network, meaning that K alternative paths are available for each node to reach each other node of the network. While this approach provides a reliable routing scheme for scenarios with high churn rates, the maintenance cost is high. The solution does not consider the heterogeneity of the uptime distribution and in the evaluation of the approach a uniform failure rate is assumed. In contrast, Omicron has been design in a way that capitalizes on the existence of the most stable nodes to greatly reduce the workload generated by highly unstable peers.

Rhea et al. [44] address the problem considering three factors: reactive versus periodic failure recovery, message timeout calculation and proximity neighbor selection. For this context, Bamboo is used as the utilized DHT infrastructure. By emulating the system, it is possible to extract critical information on the networking effects. For example, how the message timeout selection influences the process. Moreover, this work incorporates sampling as a mechanism to obtain information and select physically close neighbors. They samplings algorithms are different from these suggested for Omicron since they serve different purposes. Moreover, heterogeneity of uptime distribution is not capitalized in the way Omicron dos, since it aims in addressing approaches where each peer has been assigned equivalent roles.

The advantage of the clustering mechanism of Omicron combined with the ability of selecting the assigned

roles provide the means to utilize and benefit from the heterogeneity of the node behavior in a novel way as compared to the existing approaches. Clustering has been suggested in other pieces of work. Yang et al. [45] clusters peers as a subnetwork around *super-peers*. While one might consider an Omicron Maintainer as a super-peer, in reality Omicron differs significantly from such an approach. Omicron routes queries over a prefix-based DHT achieving efficient routing and even workload distribution. Considine [35]) suggests the usage of clusters in DHTs (i.e., Chord) to reduce the routing cost of a flat approach.

7 Conclusions

Omicron is a hybrid overlay network that has been designed to meet several critical requirements for P2P systems, which fits adequately to the needs of a great multitude of P2P based CDNs. In this paper, we have focused on the sampling mechanism of Omicron. This mechanism has been employed in providing a well balanced and stable network, though the evaluation f the stability if is not the focus of this paper. The maintenance overhead that is required to ensure network stability through cluster endurance has been evaluated with multiple sampling algorithms. Both deterministic and probabilistic algorithms have been employed where the latter showed better cluster coverage capabilities. In addition, the cluster sampling inter-arrival distributions have been estimated revealing an interesting exponential distribution property that can be further exploited by analytical means to obtain a stochastically described P2P network. The different rates of these exponential distributions reveal properties of the topology and can be used to identify potential traffic hot-spots. Moreover, the covertime by applying random walks has been investigated and approximated with that of independent sampling algorithms. The graph characteristics and properties of both Omicron and Chord allow such an approximation.

The set of mechanisms proposed in Omicron cooperate harmonically to provide a P2P overlay network meeting the large majority of the relevant requirements. For networks of small size, the long random walk mechanism will provide the best trade-off between the achieved coverage and the generated sampling traffic. As the network size grows, the short random walk algorithm is an interesting alternative.

Sampling techniques are of a more general interest, and they can be applied to other quantities than cluster endurance. In fact, they are interesting candidates in many load-balancing problems. They fit well in the distributed nature of P2P systems where no central component exists. Their exploration is vital to develop efficient systems and frameworks that can be deployed in the context of decentralized CDN infrastructures. Our future research in the area includes mechanisms that combine caching allowing to reuse sampled information for a limited amount of time, thus, considering the peer uptime distribution. This may be combined with adapting the length of random walks in order to achieve the same coverage with less traffic.

References

1. Plagemann, T., Goebel, V., Mauthe, A., Mathy, L., Turletti, T., Urvoy-Keller, G.: From Content Distribution networks to content networks—issues and challenges. Computer Communications, ENEXT Special Issue, to appear in September (2005)
2. Oram, A.: Harnessing the power of disuptive technologies. O'Reilly, Sebastopol, CA (2001)
3. Gnutella. http://www.gnutella.com (2005)
4. eDonkey2000. http://www.edonkey2000.com (2005)
5. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a scalable Peer-to-Peer lookup service for internet applications. IEEE Trans. Netw. **11**, pp. 17–32, February (2003)
6. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable Content addressable network. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 161–172. ACM Press (2001)
7. Darlagiannis, V.: Overlay network mechanisms for Peer-to-Peer systems. Ph.D. thesis, Department of Computer Science, Technische Universität Darmstadt, Germany, June (2005)
8. Cui, Y., Nahrstedt, K.: Layered Peer-to-Peer streaming. In: Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV603), pp. 162–171, June (2003)
9. Jiang, X., Dong, Y., Xu, D., Bhargava, B.: Gnustream: a P2P media streaming system prototype. In: Proceedings of the International Conference on Multimedia and Expo (ICME) 2, July (2003)
10. Hefeeda, M., Habib, A., Botev, B., Xu, D., Bhargava, B.: PROMISE: peer-to-peer media streaming using collectcast. In: Proceedings of the ACM Multimedia, November (2003)
11. Zink, M., Mauthe, A.: P2P streaming using multiple description coded video. In: Proceedings of the 30th EUROMICRO Conference, September (2004)
12. Deshpande, H., Bawa, M., Garcia-Molina, H.: Efficient topology-aware overlay network. In: Proceedings of the 1st Workshop on Hot Topics in Networks, October (2002)
13. Deshpande, H., Bawa, M., Garcia-Molina, H.: Streaming live media over a peer-to-peer network. Technical Report 2001-31, Stanford University (2001)
14. Nicolosi, A., Annapureddy, S.: P2Pcast: a peer-to-peer multicast scheme for streaming data. In: Proceedings of IRIS Student Workshop, MIT, October (2003)
15. Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A., Singh, A.: Splitstream: high-bandwidth multicast in cooperative environments. In: Proceedings of the ACM SOSP, October (2003)

16. On, G.: Quality of availability for widely distributed and replicated content stores. Ph.D. thesis, Technische Universität Darmstadt, Germany, June (2004)
17. Izal, M., Urvoy-Keller, G., Biersack, E., Felber, P., Hamra, A.A., Garces-Erice, L.: Dissecting BitTorrent: five months in a Torrent's lifetime. In: Proceedings of Passive and Active Measurements (PAM) 2004, April (2004)
18. Cherkasova, L., Lee, J.: FastReplica: efficient large file distribution within content delivery networks. In: Proceedings of the Fourth USENIX Symposium on Internet Technologies and Systems, March (2003)
19. Bustamante, F.E., Qiao, Y.: Friendships that last: peer lifespan and its role in P2P protocols. In: Proceedings of the International Workshop on Web Content Caching and Distribution, October (2003)
20. Saroiu, S., Gummadi, P.K., Gribble, S.D.: A measurement study of peer-to-peer file sharing systems. In: Proceedings of Multimedia Computing and Networking 2002 (MMCN '02) (2002)
21. Darlagiannis, V., Mauthe, A., Steinmetz, R.: Overlay design mechanisms for heterogeneous, large scale, dynamic P2P systems. J. Netw. Syst. Manag. **12**(3), 371–395 (2004)
22. de Bruijn, N.G.: A combinatorial problem. In: Proceedings of the Koninklijke Academie van Wetenschappen, pp. 758–764 (1946)
23. Hsu, F., Wei, D.: Efficient routing and sorting schemes for de Bruijn networks. IEEE Trans. Parallel. Distrib. Syst. **8**(11), 1157–1170 (1997)
24. Liu, Z., Sung, T.-Y.: Routing and transmitting problems in de Bruijn networks. IEEE Trans. Comput. **45**(9), 1056–1062 (1996)
25. Fiol, M., Yebra, L.A., de Miquel, I.A.: Line digraph iterations and the (d,k) digraph problem. IEEE Trans. Comput. **33**(5), 400–403 (1984)
26. Bridges, W., Toueg, S.: On the impossibility of directed Moore graphs. J. Comb. Theory Ser. B **29**, 339–341 (1980)
27. Fiol, M., Llado, A.: The partial line digraph technique in the design of large interconnection networks. IEEE Trans. Comput. **41**(7), 848–857 (1992)
28. Sivarajan, K., Ramaswami, R.: Lightwave networks based on de Bruijn graphs. IEEE/ACM Trans. Netw. (TON) **2**(1), 70–79 (1994)
29. Loguinov, D., Kumar, A., Rai, V., Ganesh, S.: Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience. In: Proceedings of ACM SIGCOMM'03, pp. 395–406, August (2003)
30. Bernabei, F., Simone, V.D., Gratta, L., Listanti, M.: Shuffle vs. Kautz/De Bruijn logical topologies for multihop networks: a throughput comparison. In: Proceedings of the International Broadband Communications, pp. 271–282 (1996)
31. Kaashoek, F., Karger, D.R.: Koorde: a simple degree-optimal lash table. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), February (2003)
32. Fraigniaud, P., Gauron, P.: An overview of the content-addressable network D2B. In Annual ACM Symposium on Principles of Distributed Computing, July (2003)
33. Gong, L.: Project JXTA: a technology overview. October (2002)
34. Mischke, J., Stiller, B.: Rich and scalable peer-to-peer search with SHARK. In: 5th International Workshop on Active Middleware Services (AMS 2003), June (2003)
35. Considine, J.: Cluster-based optimizations for distributed hash tables. Technical Report 2003-031, CS Department, Boston University, November (2002)
36. Darlagiannis, V., Mauthe, A., Liebau, N., Steinmetz, R.: An adaptable, role-based simulator for P2P networks. In: Proceedings of the International Conference on Modeling, Simulation and Visualization Methods, pp. 52–59, June (2004)
37. Gkantsidis, C., Mihail, M., Saberi, A.: Random walks in peer-to-peer networks. In: Proceedings of IEEE INFOCOM 2004 March (2004)
38. Gagie, T.: Large alphabets and incompresibility. Inf. Proc. Lett., 246–251 (2006)
39. Feller, W.: An introduction to probability theory and its applications, vol. II, 2nd edn. Wiley, NewYork (1971)
40. Maymounkov, P., Maziéres, D.: Kademlia: a peer-to-peer information system based on the XOR metric. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02)(2002)
41. Meer, H.D., Tutschku, K., Gia, P.T.: Dynamic operation of peer-to-peer overlay networks. Prax. Informationsverarbeitung Kommun. (PIK) **2003**(2), 65–73 (2003)
42. Stutzbach, D., Rejaie, R.: Towards a better understanding of Churn in peer-to-peer networks. Technical Report CIS-TR-04-06, Department of Computer Science, University of Oregon, November (2004)
43. Lam, S.C., Liu, H.: Salability and accuracy in a large-scale network emulator. In: Proceedings of the 2004 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, ACM Press (2004)
44. Rhea, S., Geels, D., Roscoe, T., Kubiatowicz, J.: Handling Churn in a DHT. In: Proceedings of USENIX 2004 Annual Technical Conference, pp. 127–140. June (2004)
45. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proceedings of IEEE International Conference on Data Engineering (2003)