pWeb: A P2P Web Hosting Framework

Reaz Ahmed[1], Md. Faizul Bari[1], Raouf Boutaba[1], Shihabur Rahman Chowdhury[1], Bertrand Mathieu[2], and Alexander Pokluda[1]

[1]David R. Cheriton School of Computer Science, University of Waterloo

[r5ahmed | mfbari | rboutaba | sr2chowdhury | apokluda]@uwaterloo.ca

[2]Orange Labs, Lannion, France

bertrand2.mathieu@orange-ftgroup.com

University of Waterloo Technical Report: CS-2013-12

Abstract

In recent years, we have seen a major shift in the dominant computing platforms used by Internet users. Mobile computing platforms such as smartphones are quickly becoming the primary way many Internet users connect to the Web. Furthermore, increasingly more Web content is produced and consumed by these devices. This paper introduces pWeb, a novel peer-to-peer web hosting framework that aims to keep content near its point of production and consumption. pWeb will tap into the explosion of resources offered by these mobile devices as well as other Internet connected devices such as set-top boxes, home gateways, etc. that are also sharply increasing in number. In addition to benefiting end-users, pWeb also has the potential to benefit Internet Service Providers by reducing the amount of inter-AS traffic. pWeb is supported by our industry partners, Bell Canada Enterprises, Inc. and Orange Labs, and builds on our previous work on naming, routing and searching in peer-to-peer networks.

I. Introduction

The mode of information production, dissemination and consumption is gaining a new momentum with the advent of cheaper and more powerful home entertainment devices (like set-top-boxes, home-gateways, network-attached storages, gaming consoles, *etc.*) and hand-held devices (like smart-phones, tablets, portable gaming devices, *etc.*). Combining the powerful multimedia capabilities of the hand-held devices with the persistent uptime behavior of the home entertainment devices, users can have an elevated Internet experience in a cost-effective manner.

Hand-held devices will increase dramatically in the upcoming years, which is predictable from the prominent shift of the tech industry towards hand-held device market, specially the smart phones and tablet PCs. Equipped with powerful multimedia (*e.g.*, HD video camera, audio, GPS, *etc.*) and networking (*e.g.*, Wi-fi, 4G LTE, Bluetooth, *etc.*) capabilities, these devices are generating voluminous content. These devices are contributing significantly to the popular social networking sites (*e.g.*, Facebook) and online multimedia streaming portals (*e.g.*, YouTube). As of February 2010, YouTube served 1 billion videos per day, and more interestingly, it would take 35 hours to watch the videos uploaded to YouTube every minute. Online storage and backup solutions are yet another class of Internet applications that are consistently gaining popularity. These solutions offer reliable online storage and ease of access over the Internet.

Cloud-based solutions for online storage, backup and sharing of multimedia content over the Web have a few inherent drawbacks as pointed out in [4]. First, voluminous multimedia content has to be uploaded to the cloud-stores, which generates significant amount of Internet traffic. Canadian ISP's and AS's will be facing more and more outbound traffic and hence bandwidth cost, since majority of the datacenters are located outside Canada. Second, building new datacenters will generate more pressure on the energy sector; as of February, 2009, Microsoft's data center in Quincy, Washington was consuming 48 megawatts of electricity – sufficient to power around 40,000 homes [17].

This report is a modified version of an internal project proposal for the continuation of work on pWeb written in March 2012.

Transporting huge volumes of user generated, multimedia content to distant datacenters is not a scalable solution. Rather, semi-persistent devices like, set-top boxes, home-gateways, network-attached storages (NAS) *etc.* , with network and storage capabilities and residing near multimedia content production and consumption points can be more appropriate targets for placing these contents. This strategy will greatly reduce inter-AS traffic, provide efficient access to delay sensitive multimedia content, and reduce power and resource consumption at the datacenters [**?**].

This project aims to create *pWeb* – a novel peer-to-peer (P2P) web hosting infrastructure that will transform networked, home-entertainment devices into light-weight, collaborating web servers for persistently storing and serving multimedia and web contents. In pWeb, user generated voluminous multimedia contents will be pro-actively uploaded in a nearby network location (preferably within the same LAN or at least within the same ISP) and a structured P2P mechanism will ensure Internet accessibility by tracking the original contents and their replicas. Clearly, this is a radical departure in how information would be managed compared to the existing Web. In coordination with our industrial partners, Bell Canada Enterprises Inc. and Orange Labs, this research could dramatically change the way the Web operates.

II. Problem Description: Challenges of P2P web hosting

A P2P network is fundamentally different from a client-server architecture. First, peers in a P2P system may join and leave the network frequently, while web servers are expected to remain up continuously for long periods of time. Second, shared content in a P2P system often moves from one peer to another, whereas web pages do not usually change their location within the Internet. These differences mean that state-of-the-art P2P technology cannot be used directly to create a serverless hosting system. A number of research challenges including the following must be addressed.

A. *Location independent naming*

Web documents are identified using Unified Resource Locators (URLs), which form the hyperlink structure of the World Wide Web. However, URLs are not suitable for naming P2P web objects, due to peer and content dynamism. The domain name part of a URL essentially specifies the location of a document in the Internet. However, in a P2P environment there is no guarantee of a stable location for a document. Peers get new network addresses for each session. As a result, the Domain Naming System (DNS), which maps URLs to server IP addresses, is not adequate for naming peers or content in a P2P system. Besides DNS, search engines (like Google or Bing) provide a unanimous mechanism for keyword to content mapping. They crawl the Internet hosts with fixed domain names and DNS resolvable network addresses. In our P2P Web scenario, the search engines will not be able to index peers and their contents due to the lack of a proper naming and name resolution scheme. In summary, URL based naming and hierarchical DNS lookup are not suitable for P2P web hosting.

B. *Ensuring content availability*

In contrast to web servers, the uptime of a typical Internet user is short. In the context of pWeb, it would be required for a peer to remain online round the clock to host its web contents, unless some measure is taken to host the contents during its off-line period. Contemporary P2P techniques rely on content replication to increase availability; however, they do not focus on content persistence over time. Besides the reliability requirement, security and privacy issues have to be considered while placing contents in a P2P web hosting environment.

C. Peering protocol for P2P web servers

Conventional web servers for hosting dynamic content are resource hungry. They are not suitable for deployment on low-end consumer communication or entertainment devices. In addition to hosting dynamic content, P2P web servers have to collaborate with each other for content replication. However, HyperText Transport Protocol (HTTP) - the language spoken by the web servers - is designed for client-server communication only. This protocol has to be extended for enabling P2P collaboration of web servers. A lightweight P2P client, implementing the following components is essential for successful P2P web hosting: (a) a HTTP server, (b) P2P communication protocols, (c) a trimmed down version a server-side scripting technology, and (d) a flexible, light-weight XML database system.

Based on our previous study of naming [3], [6], flexible P2P search techniques [1], [**?**], [2] and P2P availability [28], we believe that an in depth investigation of webpage and website naming in P2P environment is essential. A few research works, including [21], [15], have proposed solutions for parts of the puzzle; but a complete, working solution is still needed. We believe that an in-depth investigation is required in order to realize a comprehensive, working solution. pWeb will use a completely new architecture for serverless web hosting over structured P2P networks.

III. Research Objectives

There are a number of research challenges that must be solved in order to realize serverless web hosting in a P2P environment. Our goal is to address each of these challenges as outlined below:

- To design a location and time independent naming scheme for persistent linking of web documents in non-persistent P2P environments.
- To develop a persistent P2P substrate for peer and web content information indexing for scalable, efficient and flexible name resolution.
- To design a replication strategy for ensuring 24/7 availability of documents in dynamic P2P systems.
- To develop an open, flexible, and lightweight software architecture for hosting static and dynamic web contents at each peer with minimal resource requirements.

The long term objective is to develop and deploy a working P2P Web system, named **pWeb**, that demonstrates the viability of the serverless web hosting concept and consequently its potential for adoption at large scale.

IV. Technical Approach

In order to address the challenges and achieve the objectives discussed in Sections II and III respectively, are building architecture for pWeb based on our previous research on naming [3], P2P search [1], [2] and P2P availability [28]. Fig. 1 presents a conceptual model of the proposed architecture.

As depicted in the left part of Fig. 1, we organize the system participants in three tiers. The bottom tier (Tier-I) consists of non-persistent, portable devices like smart phone, tablets, laptops, portable gaming consoles, *etc*. These devices are the main source of web content generation as well as consumption. Consumer communication and entertainment devices, like home gateways, set-top box, network attached storage, game console and desktop computers are placed in the middle tier (Tier-II). These devices are not mobile and connect to the Internet through the same Internet Service Provider (ISP) for prolonged period of time. Generally, these devices stay online longer than the devices in Tier-I. In many cases, these devices have moderate storage capacity or provision for installing additional storage (*e.g.*, external USB drive). Finally, the persistent components like Internet servers and datacenters are placed at the topmost tier (Tier-III). Above Tier-III conventional search engines can crawl the Tier-III devices to indirectly index the web content published by Tier-I and Tier-II devices.

Right side of Fig. 1 presents the data and index flow in our system. Voluminous web content generated in a Tier-I device is replicated to a Tier-II device withing the same LAN (see edge b), while the meta information on the content is indexed at Tier-III (see edge a). Tier-II devices collaborate in small time-based replication groups to ensure 24/7 availability. Tier-II devices are given unique names, yet their network addresses are expected to change over time. Tier-III devices will retain the name to address mapping for the Tier-II devices. In addition, Tier-III devices will index meta-information on the web content stored in Tier-II devices and information on the replication groups that exist between Tier-II devices. Logically each Tier-III device will belong to separate Autonomous System (AS). Tier-III devices will collaborate using a structured P2P protocol (here Plexus) to enable efficient and distributed search of the web content hosted in Tier-II devices. Search engines can safely index the Tier-III devices to enable even faster access to the contents in Tier-II devices.

A user can search the Web as usual and can discover a content indexed at Tier-III network (Step 1 in Fig. 1). Alternatively, a user can forward a query to a Tier-II node, which can perform a distributed search on behalf of the peer using the Plexus protocol (Step 2 in Fig. 1). In either case, the user will obtain the network address of a Tier-II device which is currently alive and holding the requested content. Finally, the user can access the actual content from the Tier-II device (Step 3 in Fig. 1).

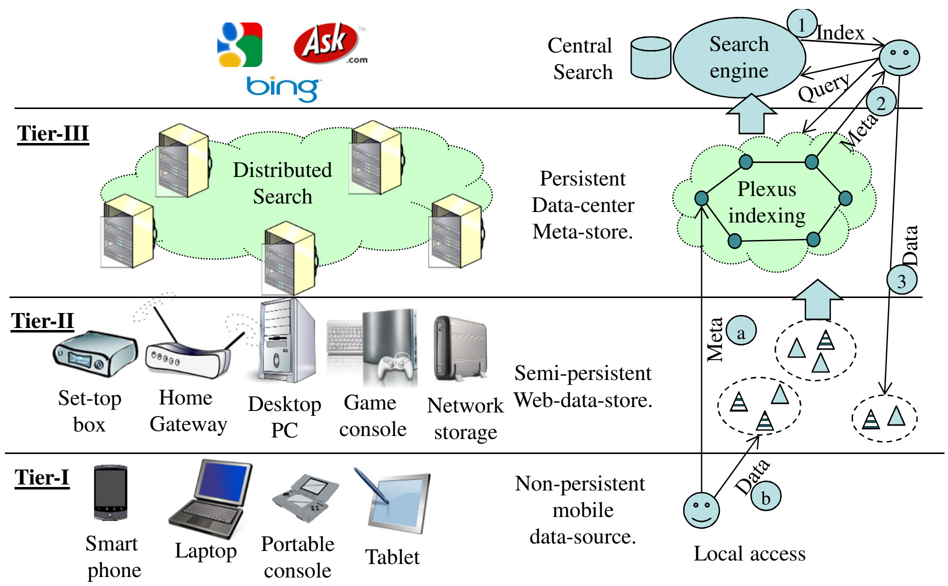


Fig. 1: Conceptual model

There are some aspects worth mentioning regarding the design choices behind this architecture. We exploit the possible physical and network proximity of the Tier-I and Tier-II devices to opportunistically upload and replicate voluminous web content from Tier-I devices to Tier-II devices. This will localize the upload traffic from Tier-I devices to Tier-II devices within a LAN. With the same spirit, we group Tier-II devices based on network proximity and complimentary uptime behavior. We intend to conceal the network traffic between Tier-II devices within individual ASs. Finally, meta-information on Tier-II web content is stored in Tier-III. Notably, volume of meta-information is negligible compared to the original web and multimedia content volume.

In the following we provide more details on the components of this architecture, namely the models for a) naming, b) indexing and c) availability.

A. Naming

A suitable naming system for P2P Web deployment should be independent of the spatial and temporal scope of the referred document. There should also exist easy conversion mechanism for converting URLs to the new naming system, and vice versa. P2P Web system requires a human-readable, flexible naming scheme. The naming authority should be distributed as well as the name resolution architecture. The naming scheme should be compatible with widely accepted Internet naming standards. None of the existing naming schemes, as investigated in our previous work [3], posses all of the above mentioned characteristics.

More recently, we proposed a naming system for P2P web hosing in [6], [**?**]. We proposed a multi-faced naming scheme, called pRL (P2P Resource Locator), which comprises three components: i) a UUID for system use, ii) a human-friendly component and iii) a set of descriptive key-value pairs. In that work we focused on content naming only. But, for a fully functional P2P Web implementation we require to name the following entities:

1) **Peer**: Peer names should be unique. A peer has to register a name with the system before using it. A returning peer will reclaim its registered name using a challenge/response mechanism.
2) **Group**: Websites will be replicated within automatically maintained small groups of peers. These groups are visible to the users. Hence group names will only have the UUID part of a name.
3) **Website**: To separate a website from the hosting peer, we will use separate namespaces for websites and peers. The originating peer's name and the replication group's UUID will be stored in the website's key value list. We will allow multiple websites to have the same name, but the UUID and key-value list will be different. For disambiguating between multiple names, the system will use the UUID part and the users will use the key-value list.
4) **Web content**: For naming webpages or multimedia content we intend to use hierarchical path names relative to the website's pRL, much like relative names in the URL scheme. In addition, security information (*e.g.*, public key, checksum *etc.*) will be indexed along with a pRL.

In addition we have to devise an efficient name resolution scheme for efficiently indexing and resolving the various types of names in the system. To this end we intend to use the Plexus protocol as explained in the next section.

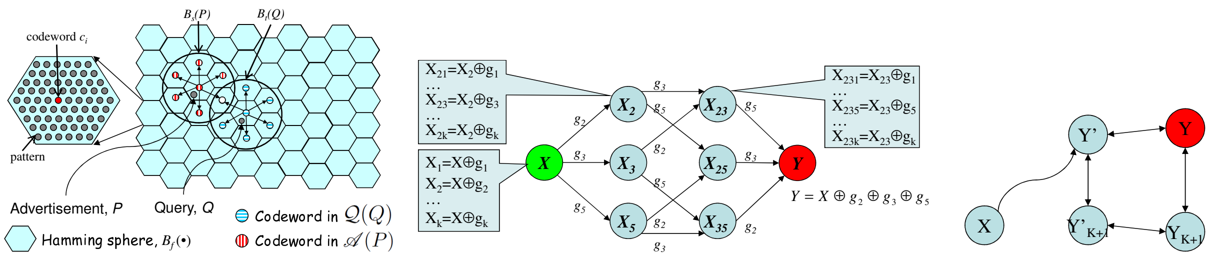
B. Distributed index and search

Distributed indexing and lookup is an integral part of the pWeb architecture for a number of reasons. First, we need to securely maintain peer-name to network address mapping and ensure efficient lookup. Second, we have to index and lookup peer and group availability information during group formation process. Third, we need to index the keywords describing the web content stored in Tier-II peers. We have previously developed a distributed search technique named Plexus [2], that supports efficient, approximate matching. Within this project, further investigation will be dedicated to enhance Plexus indexing to support the above mentioned heterogeneous indexing.

Like other DHT techniques Plexus supports efficient routing which scales logarithmically with network size. In addition, support for approximate matching is built into the Plexus routing mechanism, which is not easily achievable by other DHT techniques. To cope up with churn in P2P systems, Plexus supports multipath routing and efficient replica placement. Plexus delivers a high level of fault-resilience by using replication and redundant routing paths. Because of these advantages we propose to incorporate Plexus routing at the core of our P2P web hosting system.

In Plexus keywords are mapped to Bloom filters [8] (or bit-vectors). A Hamming distance based technique derived from the theory of *Linear Covering Codes* [11] is used for routing. The keyword to Bloom filter mapping process retains the notion of similarity between keywords, while Hamming distance based routing delivers deterministic results and efficient bandwidth usage.

In Plexus, advertisements and queries are routed to two different sets of peers in such a way that the queried set of peers and the advertised set of peers have at least one peer in common, whenever a query pattern is within a pre–specified Hamming distance of an advertised pattern. As explained in Fig. 2(a), a linear covering code ($\mathcal{C}$) partitions the entire pattern space $\mathbb{F}_2^n$ into Hamming spheres, represented by hexagons. A codeword ($c_i \in \mathcal{C}$) is selected as the unique representative for all the patterns within its Hamming sphere. To facilitate approximate matching in Plexus, an advertisement pattern, say P; is mapped to all codewords, denoted by $\mathscr{A}(P)$, that are within a pre–specified Hamming distance, say s,



(a) Hamming distance based indexing (b) Possible routing paths between peer X and Y (c) Routing under failure

Fig. 2: Core concepts in Plexus

from P. Mathematically $\mathscr{A}(P)$ can re represented as, $\mathscr{A}(P) = B_s(P) \cap \mathcal{C} = \{Y | Y \in \mathcal{C} \wedge d(Y, P) \leq s\}$, where $B_s(P)$ is the Hamming sphere of radius s centred at P and $d(Y, P) = |Y \oplus P|$ is the Hamming distance between Y and P. Similarly, a query pattern, say Q, is mapped to a set of codewords $\mathcal{Q}(Q) = B_t(Q) \cap \mathcal{C}$, for some pre–specified Hamming distance t. It is shown in [2] that there will be at–least one common codeword in $\mathscr{A}(P)$ and $\mathcal{Q}(Q)$, if $d(P, Q) \leq s + t - 2f$, where f is the covering radius of $\mathcal{C}$. In other words, by looking into the codewords in $\mathcal{Q}(Q)$, one should be able to find all advertised patterns within Hamming distance $s + t - 2f$ from Q.

***Plexus Routing**:* Consider a (n, k, d) linear covering code $\mathcal{C}$ with generator matrix $G_{\mathcal{C}} = [g_1, g_2, \ldots, g_k]^T$. To route using this code, a peer responsible for codeword X, has to maintain links to $(k + 1)$ peers with codewords $X_1, X_2, \ldots, X_{k+1}$, computed as follows:

$$X_i = \begin{cases} X \oplus g_i & 1 \leq i \leq k \\ X \oplus g_1 \oplus g_2 \oplus \ldots \oplus g_k & i = k + 1 \end{cases} \tag{1}$$

Now, the routing process in Plexus can be best explained by the example in Fig. 2(b), which shows the possible routes from peer X to peer $Y = g_2 \oplus g_3 \oplus g_3$ (any codeword Y can be generated from any other codeword X as follows: $Y = (X \oplus g_{i_1} \oplus g_{i_2} \oplus \ldots \oplus g_{i_t})$, where $g_{i_1}, g_{i_2}, \ldots g_{i_t} \in G$ and $\oplus$ is bitwise XOR operation). Peer X will forward the message to any of $X_2 (= X \oplus g_2)$, $X_3 (= X \oplus g_3)$ or $X_5 (= X \oplus g_5)$, which are one hop nearer to Y than X. If the message is forwarded to X_2 then X_2 can route the message to Y via $X_{23} (= X \oplus g_2 \oplus g_3)$ or $X_{25} (= X \oplus g_2 \oplus g_5)$. In such an overlay, it is possible to route a query from any source to any destination codeword in $\frac{k}{2}$ or fewer routing hops [2].

In Plexus protocol, a peer say Y replicates its indices to peer Y_{K+1}. In presence of failure a peer's replica can be reached in just 2 extra hops, which can be explained using the example of Fig. 2(c). Here peer X is attempting to route a query to peer Y, which has failed. When a neighbor (Y') of Y detects the failure, it forwards the query to its own replica Y'_{K+1} in one hop. Next peer Y'_{K+1} forward the query to peer Y's replica Y_{K+1} in one hop.

However, Plexus is optimized for P2P content sharing environments, which has several behavioral differences compared to the P2P web hosting scenario:

- *Replication behavior*: In a P2P network, downloaded copy of a shared content becomes a source for future downloads. While in our context, content authenticity is an important factor deciding web contents placement. A popular content may be replicated at multiple locations, but content authenticity has to be ensured.
- *Query behavior*: The number and variety of documents in the P2P Web scenario will be much higher than that in a P2P content sharing system. This will result into higher query traffic and index volume.
- *User connectivity pattern*: In pWeb, Tier-II peers are expected to host web content for longer periods of time, compared to the peers in a traditional P2P content sharing system. Tier-II peers will join and leave the network periodically, but it is expected that a Tier-II returning peer will retain the replicated contents from its previous session and will continue to host those contents.
- *Full text indexing*: In content sharing P2P systems a few keywords are advertised for each shared content and a query string comprises a subset of the advertised keywords. On the other hand,

Web search engines use many important keywords per webpage, while web queries involve a few keywords. In essence the gap between the number of keywords per advertisement and the number of keywords per query is much higher in the traditional Web scenario compared to P2P content sharing scenario.

The original Plexus routing mechanism has to be modified in order to handle the above mentioned behavioral differences between P2P web hosting and P2P content sharing. We plan to utilize the inherent capability of Plexus for trading off query traffic with advertisement traffic. As the expected advertisement rate in our case is much smaller than query rate, we can increase the number of Tier-III nodes indexing a content at Tier-II nodes, which will help in reducing the number of Tier-III nodes to be searched for query lookup. To cope with ad hoc connectivity in Tier-I and Tier-II nodes, we will assign each node a unique name. This will enable a node to host websites or contents from its previous sessions. Differential updates will be propagated to the returning peer during the rejoin process.

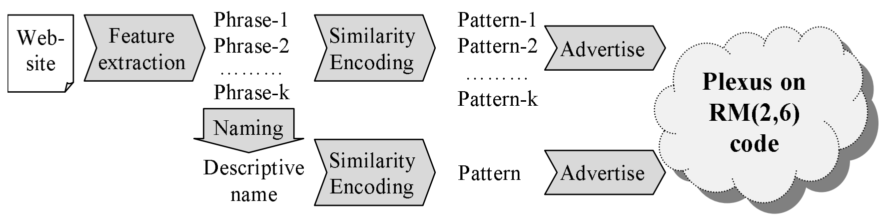


Fig. 3: Modified Plexus framework for Web document advertisement

To incorporate full text indexing capability, we will modify the advertisement mechanism in Plexus framework [2] as follows (see Figure 3).

- Instead of advertising one pattern per document we will advertise one pattern per phrase. Phrases shall be extracted by applying a feature extraction mechanism, such as Latent Semantic Indexing (LSI), over all the advertised web content on a given website. This will enable us to perform keyword search on the webpages similar to the way we search the Internet.
- For each extracted phrase we will first apply a phonetic algorithm like Soundex or Metaphone, and hash the resulting words into a Bloom Filter. There will be one Bloom filter per phrase. Use of phonetic encoding will increase the degree of similarity matching offered by Plexus.
- Since the expected edit distance between the advertised phrases and query keywords will be small, we can use a second order Reed-Muller code or a Reed-Solomon code, instead of the currently used Extended Golay code G_{24}.

C. Developing a reliable availability model

We propose to intelligently group peers so that the members of a group will remain on-line in turn and cumulatively ensure 24/7 availability of their hosted websites. Peers with complementary daily uptime cycles can be grouped together. To automate the grouping process, the following measures can be taken:

- *Gathering uptime history*: A distributed logging mechanism is required for recording peers' arrival and departure times. Such log can be maintained by using the indexing and lookup mechanisms of the Plexus framework. To model the availability pattern of a peer, we will divide a day into (say 12) time-slots and gradually compute the probability of the peer being online in each slot.
- *Forming replication groups*: Based on the uptime distribution history, we can derive a mathematical model to form locally optimal peer groups in such a way that the group size is minimal and at least one peer is always available with high probability. This group formation process should also consider hosting load at the peers and the degree of similarity in hosted content at each peer.
- *Linking to the lookup mechanism*: In order to integrate the replication groups with the Plexus indexing mechanism we have to incorporate an additional layer of lookup as follows:

- Peers having complementary uptime history will collaborate in small groups. Each group will be assigned unique GroupID. Group information like GroupID, member list, members' online status, IP address etc., will be indexed in Plexus.
- While advertising a website, the hosting peer will register its GroupID instead of its own ID with the Plexus indexing mechanism.
- While searching the network, a peer will first extract the GroupID for a query keyword. Then querying peer will lookup the currently alive peer(s) with that GroupID. Finally, the actual document will be downloaded for viewing from a live hosting peer in that replication group.

The novelty of the proposed replication scheme is that we replicate content across time, whereas existing replication schemes replicate content across on-line peers. In [28] we have presented a mechanism for improving availability in unstructured P2P systems by grouping peers having complementary diurnal availability pattern. In this work, we intend to utilize the pattern matching capability offered by Plexus to form globally optimal replication groups, which will ensure maximal availability with minimal replication overhead.

V. Related Work

P2P web hosting involves a number of research challenges. To our knowledge, there exists no research work that addresses all of these challenges as a whole. Some of these challenges have been addressed by previous research works, though not in P2P web hosting context. In the following subsections, we present the research works related to each of the issues identified in Section IV.

A. Naming

The challenge of achieving persistent names for peers and shared contents in a P2P network has not been addressed so far. Content sharing P2P systems (e.g., Gnutella, Kaaza, Morpheus etc.) use descriptive keyword list for content naming and randomly selected, temporary identifiers for peer names. Bit-torrent [10] protocol uses SHA-1 values for naming document pieces, while peers are identified by IP:Port values. In existing P2P systems peers are considered to be memoryless, i.e., peers are not assumed to retain knowledge about the overlay network from their previous sessions. Hence, the requirement for assigning persistent names to peers and content is not important in those systems. But for pWeb we have to ensure persistent names for peers, websites and web documents.

A number of research works, including [12] and [31], focused on implementing DNS lookup using P2P systems. All of these works use DHT-techniques for P2P lookup and support exact name translation. pWeb requires partial name resolution along with a way to map a name to the address of a live replica, as explained in Sections IV-A and IV-C.

B. Distributed index and search

Distributed search techniques focusing on bandwidth efficient, partial keyword search capability for content sharing P2P systems can be classified as *structured* or *unstructured* based on their indexing scheme [5]. Structured search techniques offer efficient lookup of numeric key to peer address though they lack the ability of partial keyword matching. Unstructured search techniques are capable of partial keyword matching but are inefficient in terms of bandwidth usage and are not scalable.

Few research works extend Distributed Hash Table (DHT) based systems to support *Partial matching*, e.g., Squid [26] and pSearch [33] are built on top of the DHT techniques Chord [32] and CAN [23], respectively. These techniques are not suitable for P2P Web search since they support only prefix matching and require multiple DHT-lookups to resolve a query. It has been pointed out in [20] that implementing *full text search* on a DHT network is not feasible. On the other hand, our search mechanism Plexus [2], is a DHT mechanism that utilizes Hamming distance-based routing and is inherently capable of resolving partial match queries without sacrificing efficiency. The partial match query resolving capabilities were later extended [**?**] for routing on names in Information Centric Networks [**?**].

C. Availability and content persistence

For increasing content availability in P2P systems, time-based grouping of peers is proposed in [27], [14], [16] and [18]. In [27], peers are grouped based on the probability of being online and erasure codes are used to improve content availability. In [14], a real-world trace driven approach has been proposed for unstructured networks, but no mechanism for tracking uptime statistics has been proposed. In [16], a mathematical model for measuring time-based and presence-based availability has been presented. In [18], a replication strategy for improving availability in structured P2P network has been proposed. That scheme is based on the mean-time-to-failure of a replica. In [7], [22] and [25] gossip based protocol has been used to replicate content based on tentative lifetime of the peers. In contrast to the existing research works, we propose to maintain the daily online behavior pattern of each peer in a global DHT (i.e., Plexus) and cluster peers of complementary availability pattern into small groups of 4-8 peers. Replicating content withing these small groups would ensure 24/7 availability while keeping replication overhead low.

D. P2P web hosting

To the best of our knowledge, no existing research work addresses the challenges of hosting dynamic webpages on P2P networks, which we intended to use as the database platform for dynamic page hosting. However, there has been significant research on searching and managing distributed XML databases on P2P systems. Surveys on P2P XML databases can be found in [19] and [30]. Web browsing over P2P networks is offered by FreeNet [9], FlashBack [13] and Web2Peer [24], under the assumptions that only static webpages are hosted and page replicas are independent. Webpage availability is increased through replication over the P2P network. In all of these systems, the P2P network is used for locating and caching the webpages and the Internet web servers act as the source of webpages cached in the P2P system. Our proposal differs from the above mentioned research works in two ways: firstly, we aim to achieve self-sustainable P2P web hosting without the need for persistent web servers, and secondly, we want to host dynamic webpages in the P2P environment.

References

[1] R. Ahmed and R. Boutaba. Distributed pattern matching: A key to flexible and efficient P2P search. *IEEE Journal on Selected Areas in Communications (JSAC)*, 25(1):73–83, January 2007.

[2] R. Ahmed and R. Boutaba. Distributed Pattern Matching for P2P Systems. In Proceedings of *10th IEEE/IFIP Network Operations and Management Symposium*, pp. 198–208, April 2006.

[3] R. Ahmed and R. Boutaba. Plexus: A scalable peer-to-peer protocol enabling efficient subset search. *IEEE/ACM Transactions on Networking*, 17(1):130–143, February 2009.

[4] R. Ahmed, R. Boutaba, et al.. Service naming in large-scale and multi-domain networks. *IEEE Communications Surveys*, 7(3):38–54, Third Quarter 2005.

[5] N. Shahriar, S.R. Chowdhury, et al. Ensuring β-availability in P2P Social Networks. In *5th International Workshop on Peer-to-peer Computing and Online Social Network (HotPOST)*, July 2013.

[6] M.F. Bari, S.R. Chowdhury, et al. A Survey of Naming and Routing in Information Centric Networks. *IEEE Communications Magazine*, 50(12):44–53, December 2012.

[7] R. Ahmed, M.F Bari, et al. αRoute: A Name Based Routing Scheme for Information Centric Networks. In Proccedings of *IEEE INFOCOM*, pp. 90–94, April 2013, Turin, Italy

[8] M. Armbrust et al. Above the Clouds: A Berkeley View of Cloud Computing. In *Tech. Report, EECS Department, University of California, Berkeley*, UCB/EECS-2009-28, February, 2009.

[9] S. Androutsellis-Theotokis and D. Spinellis. A survey of Peer-to-Peer content distribution technologies. *ACM Computing Surveys*, 45(2):195–205, December 2004.

[10] M. F. Bari et al. Data Center Network Virtualization: A Survey. *IEEE Communications Surveys and Tutorials*, August, 2013.

[11] M. F. Bari et al. Persistent Naming for P2P Web Hosting. *IEEE Intl. Conf. on Peer-to-Peer Computing*, September, 2011.

[12] M. F. Bari et al. A Naming Scheme for P2P Web Hosting. *IEEE Journal on Selected Areas in Communications*, September, 2013.

[13] S. Blond, F. Fessant, and E. Merrer. Finding good partners in availability-aware p2p networks. In *Proc. 11th Int. Symposium on Stabilization, Safety, and Security of Distributed Systems(SSS)*, 2009.

[14] B H. Bloom. Space/time trade-offs in hash coding with allowable errors. In *Communications of ACM*, 13:7 422–426, 1970.

[15] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science (LNCS)*, 2009:46–66, 2001.

[16] B. Cohen. Bittorrent protocol specification. http://www.bitconjurer.org/BitTorrent/protocol.html.

[17] G. Cohen. Covering codes. North Holland Publishers, 1997.

[18] R. Cox, A. Muthitacharoen, and R. T. Morris. Serving dns using a peer-to-peer lookup service. In *LNCS: Peer-to-Peer Systems*, volume 2429/2002, pages 155–165. Springer, Jan 2002.

[19] M. Deshpande, A. Amit, M. Chang, N. Venkatasubramanian, and S. Mehrotra. Flashback: A peer-to-peer web server for flash crowds. *Proc. Int. Conf. on Distributed Computing Systems*, 2007.

[20] R. J. Dunn, J. Zahorjan, S. D. Gribble, and H. M. Levy. Presence-based availability and p2p systems. *Proc. Int. Conference on Peer-to-Peer Computing*, pages 209–216, Sept. 2005.

[21] R. J. B. Jr., A. Crainiceanu, and R. Agrawal. Peer-to-peer sharing of Web applications. In *Proceedings of World Wide Web Conference*, May 2003.

[22] O. B. Karimi et al.. Availability in peer to peer management networks. In *LNCS: Challenges for Next Generation Network Operations and Service Management*, Vol. 5297, pp. 552-555, Oct. 2008.

[23] R. H. Katz. Tech titans building boom. In *IEEE Spectrum*, February, 2009.

[24] K. Kim. Time-related replication for p2p storage system. In *Proceedings of International Conference on Networking*, pages 351–356, Apr. 2008.

[25] G. Koloniari et al. P2P management of xml data: Issues & research challenges. *SIGMOD Record*.

[26] J. Li, B. Thau, L. Joseph, M. Hellerstein, and M. F. Kaashoek. On the feasibility of peer-to-peer Web indexing and search. In *IPTPS '03*, pages 207–215, 2003.

[27] J. Li and C. Zhang. Distributed hosting of web content with erasure coding and unequal weight assignment. In *Proceedings of IEEE International Conference on multimedia and expo*, June 2004.

[28] R. Narendula, T. G. Papaioannou, and K. Aberer. My3: A highly-available P2P-based online social network. In *Proc. of IEEE P2P*, 2011.

[29] S. Ratnasamy et al. A scalable content-addressable network. In *Proc. of ACM SIGCOMM*, 2001.

[30] H. B. Ribeiro et al. Implementing a peer-to-peer web browser for publishing and searching web pages on internet. In *Proc. of Int. Conf. on Advanced Information Networking and Applications*, pp. 754-761, 2007.

[31] J. Sacha et al. Discovery of stable peers in a self-organising peer-to-peer gradient topology. In *Proc. IFIP Int. Conference on Distributed Applications and Interoperable Systems*, 2006.

[32] C. Schmidt and M. Parashar. Enabling flexible queries with guarantees in P2P systems. *IEEE Internet Computing*, 8(3):19–26, June 2004.

[33] T. Schwarz, Q. Xin, and E. Miller. Availability in global peer-to-peer storage systems. In *Proceedings of the 2nd IPTPS*, July 2004.

[34] N. Shahriar et al. Diurnal availability for peer-to-peer systems. In *Proceedings of Consumer Communications and Networking Conference*, January, 2012.

[35] S. Shi, J. Yu, G. Yang, and D. Wang. Distributed page ranking in structured p2p networks. In *Proceedings of International Conference on Parallel Processing*, pages 179–186, Oct 2003.
[36] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. pages 146–171. 2005.
[37] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. *SIGCOMM Comput. Commun. Rev.*, 32(4):73–86, 2002.
[38] I. Stoica et al. Chord: a scalable Peer-to-Peer lookup protocol for Internet applications. *IEEE/ACM Transaction on Networking (TON)*, 11(1):17–32, 2003.
[39] C. Tang, Z. Xu, and M. Mahalingam. pSearch: information retrieval in structured overlays. *ACM SIGCOMM Computer Communication Review*, 33(1):89–94, 2003.
[40] J. Wang et al. Distributed collaborative filtering for peer-to-peer file sharing systems. In *Proceedings of the ACM symposium on Applied Computing*, pages 1026–1030, 2006.