This paper deals with web caching and how this can be improved using a p2p system that interconnects the browser cache of each client with the cache of the web proxy so that clients and proxy share their cache contents. In this way hit ratio is improved within the whole system and redundant replication is avoided.

The most common caching model is that each client has its own browser cache and a group of such clients is connected to a proxy. When a request is issued, local cache is used, then the proxy cache and if still no match has been found then request is forwarded to other caches or to the original server. The answer is stored in the proxy and returned to the browser that will also store the document in its cache. This model has many drawbacks including document duplication among different cache levels, waste of storage space, limited scalability and more. The main idea in this paper is that it is possible that the requested document exists in another browser cache and can be retrieved from there. This is the case where some documents may have been replaced in the proxy cache but not in a browser cache.

The hit ratios in web proxies tend to decrease because of the rapid increase in the number of web servers and the diversity of documents that can be found today. Another reason is the prevalence of dynamic pages, which are noncacheable. A common solution is to enlarge the proxy cache size but this is not efficient and cannot keep up with the growth of the web. Experiments have shown that from 1995 until 1999 hit ration has dropped from 64% to 44%. They have found out that the storage space at the proxy should grow about 10 times in order to maintain the same hit ratio. It is clear that proxy cache size enlargement will no longer be sufficient.

The authors have also analyzed web traces in order to find how document duplication influences caching. Results have shown that about half of the document are requested only once but using the classic model for caching, these documents will remain in both proxy and browser cache. On the other hand 35% of the documents are requested from individual clients only but these documents will also be stored in the proxy cache although other clients does not need them. Finally 64% of the documents are shared among many clients but they will be duplicated in every client browser cache thus wasting storage space. Apart from that, duplication introduces overhead in order to invalidate cache contents or broadcasting changes.

The proposed architecture presented in the paper is that the proxy keeps track of client cache contents. So when there is a cache miss, the requested document is searched in other client caches. If there is a hit, then the document is forwarded to the client who made the request and stored in the appropriate cache in order to reduce document duplication. Specifically upon a client request one of the following scenarios may occur:

1. Browser cache hit: Document is read from there
2. Proxy hit: The document has been found in the proxy cache. The access counter for this document for this client is increased and data are returned to the client. If the counter is above a specified threshold (meaning that this document is frequently requested from this client), the client is informed to cache the document in its local cache
3. Proxy's browser index file hit: The document has been found in another client's cache and is forwarded to the client that made the request. Two counters for the specified document are updated: one global, containing all requests, and one for the specific client that requested the document. If the first one is above a threshold (meaning that the document is accessed from multiple clients), the document is moved to the proxy cache. If the second counter is above a threshold (meaning that is frequently accessed from another client), the document is cached also in the other client's browser cache
4. Global miss: No match has been found. Document is fetched from the original server (or another level of caches), forwarded to the client and cached only at the browser level.

Evaluation of current web proxies has shown that the average proxy cache size per client is relatively small. Memory space ranges from 0.04 to 0.08 MB and cache size from 7.34 to 10.86 MB, which are not large enough in practice for today's computer systems.

The authors of the paper have evaluated their system and compare it with other three models: the classic caching model, the proposed model but without methods in order to avoid document duplication and an offline algorithm that are close to optimal performance. Experiments have shown that their algorithm achieves better hit ratio and byte ratio that the classic model and the one without document duplication avoidance. Moreover, they have shown that their model will be more performance beneficial as Web servers and Web client populations continue to increase in both numbers and types. Finally experiments show that their model achieves average latency reduction of 21 percent compared to the model without document duplication avoidance and about 56 percent compared with the classic model.

Two issues that arise from this work is reliability and privacy. The first one arises from the data integrity of the data that each client has in its own cache. In order to ensure that data have not been modified, encrypted digital watermarks are used for each document. As far as privacy is concerned the system should hide client requests and identities. So the proxy acts as an anonymizer that receives request and forward in behalf of clients.