Abstract:
• Cooperative caching techniques share clients' caches to provide an efficient shared cache.
• As new blocks are read from server disk locally, cached blocks are replaced.
• This work examines existing algorithms and proposes new algorithms for cache replacement.

Background:
• Cooperative caching techniques try to minimize duplication across client caches.
• Caching algorithms thus tend to favor singlets - blocks cached by a single client.
• Existing work examines the handling of singlets.

Simulation:
• A network simulation was implemented in Java to study different replacement algorithms.
• Simulation parameters included number of clients, cache size, total blocks, total requests.

Conclusions:
• Among algorithms using LRU cache, Robin Hood had better hit rates than N-Chance which showed better results than Greedy Forwarding.
• Algorithms replacing the most cached local block showed significantly higher hit rate, but shared cache maintenance is expensive.
• Forwarding techniques made little difference overall for targeted replacement, targeted forwarding was slightly better than random, which was just better than no forwarding.
• K-Sequence algorithm can have better hit rates for known sequences of data, but performs worse in the general case.

Future work:
• Since targeting entails greater processing work, scalability of targeted replacement vs. random forwarding could be explored.
• Investigation of automatic selection of replacement algorithm depending on the nature of network activity.
• Application of replacement techniques within a known context such as a content delivery network.
• Comparative study of replacement techniques employing probabilistic data structures such as Bloom filters with those studies here.