21

CONTENT REPLICATION AND DELIVERY IN INFORMATION-CENTRIC NETWORKS

Vasilis Sourlas[1], Paris Flegkas[1], Dimitrios Katsaros[1], and Leandros Tassiulas[1]

[1] *University of Thessaly, Oktovriou, Volos, Greece*

21.1 INTRODUCTION

Information-centric networking (ICN) is emerging as the main future networking environment, given that the vast majority of Internet activities are related to information access and delivery. In ICN, information is explicitly labeled so that anybody who has relevant information can potentially participate in the fulfillment of requests for said information. Publications are issued by clients (publishers) when they have a new information item to publish in the network, while subscriptions are issued by clients (subscribers) to subscribe the items they are interested in. Given the information-centric nature of the distribution utilizing information that is replicated across almost ubiquitously, available storage devices are an almost natural thought. Optimized dissemination of information within transient communication relationships of endpoints is the main promise of such efforts, and efficient replication of information is key to delivering on this promise.

While packet-level in-network opportunistic caching is one of the salient characteristics of ICN architectures, proper cache placement and replica assignment still have an important role to play. Content delivery network (CDN)-like replication distributes a

Advanced Content Delivery, Streaming, and Cloud Services, First Edition.
Edited by Mukaddim Pathan, Ramesh K. Sitaraman, and Dom Robinson.

site's content across multiple mirror servers. When a client is interested in a particular piece of information, his/her request is redirected to one of the existing replication points rather than requiring retrieval from the original publisher. Replication is used to increase availability and fault tolerance, while it has as side-effect load-balancing and enhanced publisher subscriber proximity. Particularly, CDN providers strategically place surrogate servers and connect them to Internet service provider (ISP) network edges so that content can be closer to clients. Given the significant impact that content delivery has on the utilization of an ISP network, some work has recently started to investigate new models and frameworks to support the interaction between ISPs and CDNs.

CDNs and ICN are designed to fulfill the necessity of efficient content delivery. Their main difference is that CDNs build up the end-to-end content delivery in the Internet at the application layer, while ICN is a clean slate proposal for an alternative approach to the core architecture of the network. In this chapter, we present a three-phase framework as a contribution to the problem of information replication in an ICN environment, through the synergy of ICN with CDN techniques. Moreover, we believe that this synergy will enable CDNs to incorporate, through the ICN functional components, dynamic network information on replica selection to determine the best paths over which transfer of content will take place. The objective of the presented framework is to minimize the total traffic load in the network subject to installing a predefined number of replication devices, and given that each device has storage limitations. The presented framework comprises three phases, namely, the *Planning*, the *Offline Assignment*, and the *Online Replacement* phases, which manage the content and the location of each replication device in the network.

In the planning phase, the presented framework selects those nodes of the network to place the replication devices (CDN server) while in the offline assignment phase each information item is assigned, based on its popularity, at a subset of the selected replication points so that the targeted objective is satisfied. Finally, the online replacement/reassignment phase dynamically reassigns information items in the replication devices based on the observed items' changing request patterns. In order to support the presented framework, the three phases should be provided by relevant functional components. Regarding the Planning and the Assignment phases, these components reside outside the network and run offline algorithms at two different but long-medium timescales, while the replacement phase is residing in components installed at each replication device of the network and run in real-time scale. The three-phase replication framework is generic so that it can apply in almost every ICN proposed architecture.

The rest of this chapter is organized as follows. In Section 21.2, a brief related work on ICN architectures and replication is given. In Section 21.3, we present the three-phase replication framework, whereas in Section 21.4, we shortly evaluate the presented algorithms. Section 21.5 is devoted in future research directions, whereas in Section 21.6 we conclude this chapter.

21.2 RELATED WORK

ICN is a flexible communication model that meets the requirements of the information distribution in the Internet, since information is addressed by semantic attributes rather than origin and destination identities. In recent research efforts, among others named data networking/content-centric networking (NDN/CCN) [1] and Publish Subscribe Internet Technology (PURSUIT) [2] aim to switch from host-oriented to content-oriented networking by naming data/content instead of naming hosts in order to achieve scalability, security, and performance.

NDN [1] proposes a name-based routing system for locating and delivering named data packets. The fundamental entities in NDN are Interest and Data packets. When a user wishes to receive data, he/she issues an Interest that contains the data name. The network propagates the Interests to the nearest data source (anycast), and then the requested item is delivered back to the user in the form of a Data packet. NDN uses names to identify content objects only; there is no notion of host name, point of attachment, or path identifier. Content names follow a hierarchical form similar to URLs or file system paths, and by definition, Interest and Data paths are symmetric.

In PURSUIT [2], the design paradigm involves three separate elements and three separate functions: publishers, subscribers, and the REndezvous NEtwork (RENE) on the one hand, and the functions of rendezvous, topology management/formation, and forwarding on the other hand, respectively. While the first three elements also exist in other candidate architectures under different names, the design principle of PURSUIT is to clearly distinguish the latter three functions. In more detail, publishers in PURSUIT advertise the availability of information by issuing a publication message to the RENE. Similarly, subscribers are entities interested in consuming information who express their desire by issuing a subscription message to the RENE for a specific piece of information. The RENE is responsible to match publications to subscriptions through the rendezvous function and choose the best route (through the topology management and the forwarding functions) for the delivery of the requested information.

In the area of replication in ICN in Reference 3, a historic data retrieval publish/subscribe system is proposed, where databases are connected to various network nodes, each associated with a set of items to store. In Reference 3, every information item is stored only once and no placement strategies have been examined. In Reference 4, a set of offline storage planning and replica assignment algorithms for the ICN paradigm are presented, while in References 5, 6 an online approach for the reassignment of items within the replication points is presented.

In the traditional context of CDNs, the placement problem is a thoroughly investigated problem. Particularly in References 7, 8, authors approached the placement problem with the assumption that the underlying network topologies are trees. This simple approach allows the authors to develop optimal algorithms, but they consider the problem of placing replicas for only one origin server. The placement problem is in fact a numeristic polynomial (NP)-hard problem [9] when striving for optimality, but there are a number of studies [10–15] where an approximate solution is pursued. Their work is also known as network location or partitioning and involves the optimal placement of k service facilities in a network of N nodes targeting the minimization of a given objective. In some cases, it can be shown that this problem reduces to the well-known *k-median* problem.

The authors of Reference 16 model replica assignment as a distributed selfish replication (DSR) game in the context of distributed replication groups (DRGs). Under the

DRG abstraction, nodes utilize their storage to replicate information items and make them available to local and remote users. The pairwise distance of the nodes is assumed to be equal, while our framework considers the generic case of arbitrary distances. In the context of DRG and under the same distance assumption, a two-approximation cache management algorithm is presented in Reference 17. Finally, in Reference 18, authors develop a cache management algorithm aimed at maximizing the traffic volume served from the caches and minimizing the bandwidth cost. They focus on a cluster of distributed storages, either connected directly or via a parent node, and formulate the content placement problem as a linear program in order to benchmark the globally optimal performance.

More placement algorithms have been presented in Reference 9. Particularly, authors formulate the problem as a combinatorial optimization problem and show that the best results are obtained with heuristics that have all the stores cooperating in making the replication decisions. Moreover, in Reference 19, authors introduce a framework for evaluating placement algorithms. They classify and qualitatively compare placement algorithms using a generic set of primitives that capture several objectives and near-optimal solutions. In most of the above approaches, a similar cost function (optimize bandwidth and/or storage usage costs for a given demand pattern) is considered. Less attention has been given though to network constraints (limited storage capacity) and the possibility of reassigning items between the replication points as popularity and locality of users demand change.

Finally, in the research area of investigating new models and frameworks to support the interaction between ISPs and CDNs, in Reference 20, the authors highlight that CDN providers and ISPs can indirectly influence each other, by performing server selection and traffic engineering operations, respectively, and they investigate different models of cooperation between the two entities. In Reference 21, the authors propose a framework to support joint decisions between a CDN and an ISP with respect to the server selection process. This framework allows the ISP and the CDN to collaborate by exchanging some local information (network utilization from the ISP side and server conditions from the CDN side), so that it can result in better control of the resources. An ISP-supported CDN service has been presented in References 22, 23, whereby content is stored and served from within ISP domains. This solution, however, can incur high operational costs, given that ISPs will have to maintain large storage capacities, and may thus be economically unviable.

21.3 FRAMEWORK FOR INFORMATION REPLICATION IN ICN

In this section, we present separately the three phases of the framework for the management of the information replication in ICN.

21.3.1 The Planning Phase

The planning phase takes the number of available replication devices an operator wishes to install as an input, the network topology and a long-term prediction of subscriptions in the network. It can run periodically deciding the optimal placement of the replication points at a long-term timescale (e.g., once a year) or whenever the current location of them leads to an inefficient deployment due to significant subscriptions' changes not successfully predicted. Performing and enforcing the decision of the planning component usually involve high level business decisions as there is a high cost associated with moving a replication device to a different physical location or extending their number. In Reference 24, an ICN-oriented planning algorithm is presented for the selection of the replication points in the network based on the local demand for each item and the storage limitations of each replication device.

21.3.2 The Offline Assignment Phase

The offline assignment phase also runs periodically but at a medium- or long-term scale. It takes as input the outcome of the planning phase regarding the locations of the replication devices installed in the network, the physical network topology, and the medium- or long-term forecast. Replicas' relocation can be enforced by instructing the replication devices to subscribe to a different set of information items. The instruction itself is realized as a publication of an information item to which replication devices are subscribed. In general, the replication points act both as publishers and subscribers for the information items they are instructed to store. They subscribe in order to receive new versions of the items, while they act as publishers for the same items to interested subscribers. This way, when a client subscribes to a specific piece of information, one or more publishers/replication points are enabled, based on the operator's policy, to publish the relevant data. Figure 21.1 illustrates the basic modules of the planning and the offline assignment phases.

Before the presentation of the online replacement/reassignment phase, we present below an ICN-oriented planning algorithm for the selection of the replication points in the network based on the local demand for each item and the storage limitations of each replication device. We also present a mechanism for the offline assignment of the replicas of each item in the selected replication points.

21.3.2.1 Modified Greedy Algorithm We use algorithms presented in the context of CDN networks as the base of our planning and offline replica assignment scheme. Particularly in References 9, 10, authors developed several placement algorithms that use workload information, such as latency (distance from the storage points) and request rates, to make the placement decision. Their main conclusion is that the so-called greedy algorithm that places replication devices in the network based on both a distance metric and request load performs the best and is very close to the optimal solution.

The traditional greedy algorithm assumes that there exists only one class of content in the system, or equivalently there is no distinction in the content. We let r_i be the demand (in requests/s) from clients attached to node i. We also let p_{ij} be the percentage of the overall request demand accessing the target server j (traditional placement algorithms replicate a specific origin server) that passes through node i. Also, we denote the propagation delay (hops) from node i to the target server j as d_{ij}. If a replica is placed at

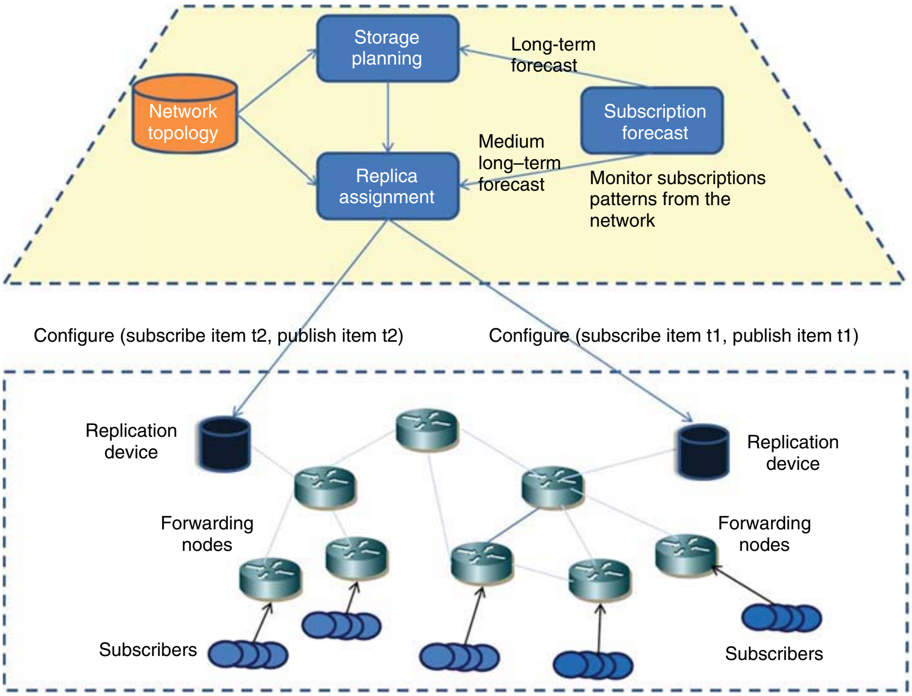

Figure 21.1 Architectural illustration of the planning and the offline assignment phases.

node i, we define the gain to be $g_{ij} = p_{ij} \cdot d_{ij}$. This means that p_{ij} percentage of the traffic would not need to traverse the distance from node i to server j.

The greedy algorithm chooses one replication point at a time. In the first round, it evaluates each node of the network to determine its suitability to become a replication point of the origin server. It computes the gain associated with each node and selects the one that maximizes the gain. In the second round, searches for a second replication point which in conjunction with the one already picked yields the highest gain. The greedy algorithm iterates until all replication points have been chosen to replicate the given server.

In many ICN implementations, the notion of an origin server usually does not exist. Publishers join the network, publish their content, and disappear. So, in order to obtain the location of the replication points, we modify the greedy algorithm. Particularly, we repeat the above-mentioned procedure as many times as the number of the nodes in the network, assuming each time that the targeted server is a different node of the network. We get in that way a set of different possible replication points per iteration. Finally, we select as our final replication points those nodes that appeared more times in the per-element summation of the different sets. The modified greedy algorithm presented here assumes uniform distribution of the probability among the different nodes of the network that publications could occur. Of course, other forms of probability distributions could be used, and each different set should be first weighted with its probability before the per-element summation.

21.3.2.2 Planning and Offline Replica Assignment Algorithm for ICNs

Here, we use the modified greedy algorithm described previously for the case where not one but M different information items in our network exist. The presented algorithm is composed of the following steps:

Step 1. For each item $m \in M$, we execute the modified greedy algorithm and we get M sets of possible replication points S_m.

Step 2. Each vector S_m is weighted by $w_m = \sum_{i=1}^{N} r_i^m / \sum_{m=1}^{M} \sum_{i=1}^{N} r_i^m$, where N is the number of the nodes in the network and r_i^m is the number of requests per second generated at node $i, (i \in N)$ for item $m, (m \in M)$. w_m is the significance of item m (popularity) regarding the traffic demand of each item in the network.

Step 3. We select as our replication points those nodes (as many as the network operator/CDN provider is willing to install) that appeared more times in the per-element weighted summation of S_m vectors. We call that vector the *replication nodes vector* S.

Step 4. For each item m, starting from the most significant (based on the weight), we assign replicas (say k_m) following the procedure below:

For each entry in S_m of item m calculated in Step 1, assign a replica if that entry also appears in S, calculated in Step 3, and only if that replication point has been assigned less than C items (storage capacity of each replication point) until we get k_m replicas (replication degree of item m, which is relative to the weight of each item in the network).

Steps 1–3 of the presented algorithm described previously comprise the planning phase of the algorithm while Step 4 is the assignment phase. Step 4 is also known as the generalized assignment problem, which even in its simplest form is reduced to the NP-complete multiple knapsack problem. For the solution of the assignment problem, we used the heuristic approaches described earlier, while more approaches could be found in the literature (e.g., References 24, 25).

21.3.3 The Online Replacement Phase

As the provisioning periods of the planning and the offline assignment phases can be quite long, the subscription patterns may significantly vary during that period. For this reason, we introduce the online replacement/reassignment phase that enables the replacement of information items to the replication points to take place in real time, based on the changing demand patterns of the users. Distributed components of that phase decide the items every replication point stores by forming a substrate that can be organized either in a hierarchical manner for scalability reasons or in a peer-to-peer organizational structure. Communication of information related to request rates,

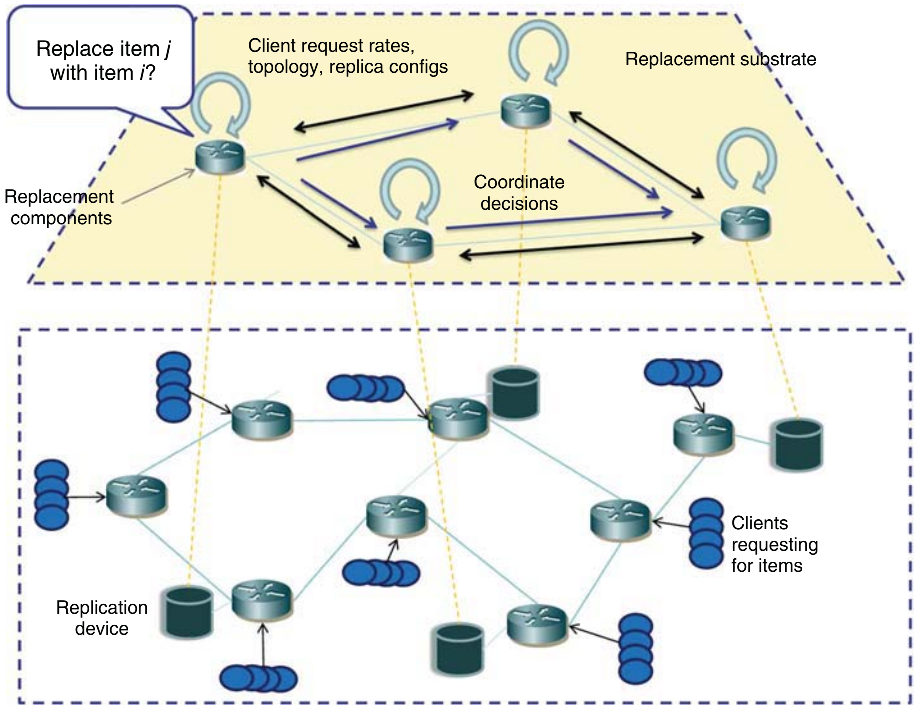

Figure 21.2 Architectural illustration of the online replacement phase.

popularity/locality of information items, and current replication points' configuration takes place between the distributed replacement components through an intelligent substrate.

Every replacement component, as depicted in Figure 21.2, should decide in a coordinated manner with other components whether to store an item. This may require the replacement of an already stored item, depending on the available space. The decision of this replacement of stored items is performed toward maximizing an overall network-wide utility function (e.g., the gain in network traffic), which means every node should calculate the gain the replacement of an item would incur. This approach assumes that every component has a holistic network-wide view of all the replication points' configuration and relevant request patterns, and this information should be exchanged periodically or in an event-based manner when a component changes the configuration of its attached replication device.

Since all the above decisions are made in a distributed manner, uncoordinated decisions could lead to suboptimal and inconsistent configurations. Coordinated decision-making of a distributed solution can be achieved through the substrate mechanisms, by ensuring that components change the configuration of each replication device in an iterative manner, that is, one at a time and not autonomously in a potentially conflicting manner.

Any distributed online mechanism that applies in this third phase should capture the volatile environment under consideration. All of them should be adaptive to popularity and locality changes by fetching new items at a replication device and replacing existing items. We envision two classes of algorithms that could be applied in the online replacement phase that differ in the amount of information that needs to be communicated through the substrate and the required level of coordination among the components. We present them in order of decreasing complexity, in terms of the induced communication overhead.

The first class, henceforth called *cooperative*, aims at minimizing the overall network traffic. This requires that every component needs a holistic network-wide view of the request patterns and the current replication points' configuration. In addition, since each replacement decision affects the whole network, cooperation in the decision-making is required. The cooperative algorithm requires at each iteration each component of the network to compute the relative gain of every possible replacement and through appropriate message exchange to cooperate for the final replacements. In other words, every component participates in the execution of the algorithm at each iteration, but every time only one (the one with the maximum relative gain) performs valid replacements.

The second class, henceforth called *holistic*, also aims at minimizing the overall network traffic and hence requires the same amount of information. However, in the holistic class, there is no need for coordination of the actions of the components, and the required decisions are made in an autonomous manner by each one individually. The holistic class is of similar nature as the cooperative and toward the same objective. Its distinguishing characteristic though is that each component operates in its own performing replacements on the respective replication point. Particularly, only one component at each iteration performs valid and beneficial replacements toward a specific objective. In both cooperative and holistic classes, it can be shown that since any change performed in the replication points' configuration decreases the overall network traffic, the presented algorithms finally converges to a stationary point where no further improvement is possible. The algorithms do not necessarily converge to the optimal assignment but to a local minimum of the objective given the initial configuration. In the presented algorithmic classes, the replacements are based on the real-time observed items' request patterns such as their popularity and locality and not in static offline predictions. In the next section, we present in detail the steps of the cooperative algorithm, and based on them we also describe the functionality of the holistic algorithm.

21.3.3.1 Cooperative and Holistic Online Replacement Algorithms

Since we wish to minimize the overall network traffic cost, we assume that the underlying content delivery mechanism always directs the requests to the closest replication point out of those holding the requested item. Given such an access mechanism, the replacement components have to coordinate their actions toward finding the replication degree and the location where each item should be cached. We assume equal capacity among the replication devices for ease of presentation, and we also assume that each information item is of unit size, which is a typical assumption in the literature (e.g., [16,17]). The presented algorithms are also applicable in the case of different item sizes as well. However, special care needs to be given, since several items may need to be removed from the replication device in order to fit the new item.

At the cooperative algorithm at each iteration, all the replacement components of the network execute the following steps in parallel, given the current storage configuration H and the corresponding total traffic cost $T(H)$.

Step 1. Let C_v denote the set of items that are stored at replication point v and in at least one more replication point in the network. For each item $m \in C_v$ compute the overall performance loss, $l_v^m = T(\overline{H}_v^m) - T(H) \geq 0$, that will be caused if item m is removed from v, leading to a valid new configuration $\overline{H}_v^m$. In this case, all the requests for item m at v will be served by another replication point, which is at least that far.

Step 2. Let P_v denote the set of items that are not stored at v. For each item $m \in P_v$, compute the overall performance gain $g_v^m = T(H) - T(\underline{H}_v^m) \geq 0$ achieved if item m is inserted at replication point v, hence leading to a new configuration $\underline{H}_v^m$. In this case, a certain amount of requests for item m will be served by node v, as the closest replica.

Step 3. Each replacement component v considers as candidate for insertion the item $i \in P_v$ of maximum performance gain and as candidate for replacement the item $j \in C_v$ of minimum performance.

Step 4. Each replacement component at replication point v calculates the maximum local relative gain $b_v = g_i - l_j$ and informs the rest of the replacement components through a report message Rep(b, v, i, j).

Step 5. After receiving the Rep messages, each replacement component calculates the most network-wide beneficial replacement, say Rep $* (b^*, v^*, i^*, j^*)$, the one of maximum relative gain, and updates its configuration matrix H (the matrix that shows where every item is stored in the replication points of the network). At this point, only the configuration matrices of the replacement components are updated. Once the algorithm has converged, these components fetch and cache new information items and replace cached ones (e.g., fetch item i^* and replace item j^*).

Step 6. Repeat Steps 1–5 until no further replacements are beneficial for the network, that is, no positive relative gain exists.

The holistic algorithm is of similar nature and toward the same objective. Its distinguishing characteristic though is that each replacement component operates on its own by performing replacements on the respective replication point. At each iteration, a single component, say v, autonomously decides and executes the following steps. Steps 1–3 are identical to the cooperative algorithm and are omitted:

Step 7. The replacement of maximum relative gain $b = g_i - l_j$ is performed by the component v. The rest of the components are notified through the report message Rep(b, v, i, j).

Step 8. After receiving the Rep message, every component updates its configuration matrix H.

Although the replacements may be applied asynchronously among the replication points, we assume that only a single component may modify the storage configuration at a given time. This is due to the requirement that each component should know the current storage configuration of the network, in order to calculate the gain and loss metrics. Thus, each modification is advertised to the rest replacement components. Relaxing this assumption would lead to a setting where the replacement components make decisions based on outdated information, causing thus some performance degradation and making convergence questionable.

21.4 PERFORMANCE EVALUATION

Here, we evaluate through simulations the performance of the presented framework and the corresponding algorithms. Since no ICN infrastructure has been deployed for commercial use yet, no publicly available datasets exist for performance evaluation. Thus, realistic synthetic workload generators are used instead. The request rate for an item at each node is determined by its popularity. Here, we approximate the popularity of the items by a Zipf law of exponent z_{pop}. Literature provides ample evidence that the file popularity in the Internet follows such a distribution [26–29]. In particular, we consider seven typical values for z_{pop} ranging from −1 to 1. We also assume that in each node of the network, a total of 200 requests per second are generated. Thus, the request rate of each item at each node varies from 0 to 200 requests/s according to its popularity.

We run two sets of experiments: one evaluating both the planning and the offline assignment phases of the presented framework, and the other evaluating only the online replacement phase after an initial planning. We used topologies from the Internet Topology Zoo dataset [30], which contains real network topologies from all over the world.

Figure 21.3 presents the performance regarding the overall network traffic (in responses × hops/s) of the planning and the offline assignment phase. By default, unless explicitly mentioned, we assume that $S = N/8$ replication points are placed in a network of N nodes, and the storage capacity of each replication point is $C = M/4$ information items. We compare the presented offline assignment mechanism with a totally random assignment procedure when we vary the size of the network, the capacity of each replication point, and the number of replication points.

From Figure 21.3, it is obvious that the presented assignment mechanism combined with the presented planning scheme performs on average 10–30% better than the random assignment. For a more detailed evaluation of the planning and the offline assignment phases, refer to References 4, 24, 31, where we observed that in the real world where a storage provider has limitations in the number of replicas that can install, each replication device has storage limitations that the presented scheme is an appropriate solution in almost any scenario.

In Figure 21.4, we investigate the adaptability of the presented online replacement algorithms as the popularity of the demand patterns change. Using as initial storage planning and assignment configuration for a given set of popularity values, we depict the performance of each algorithm as it adapts to the new environmental parameters.

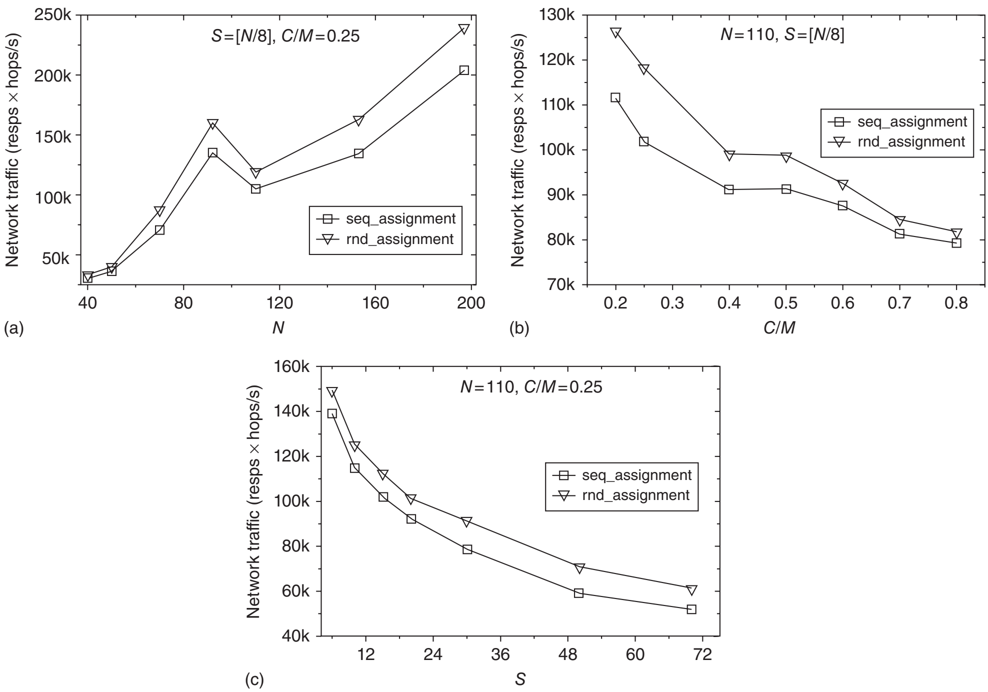

Figure 21.3 Performance of the planning and the offline assignment phase.

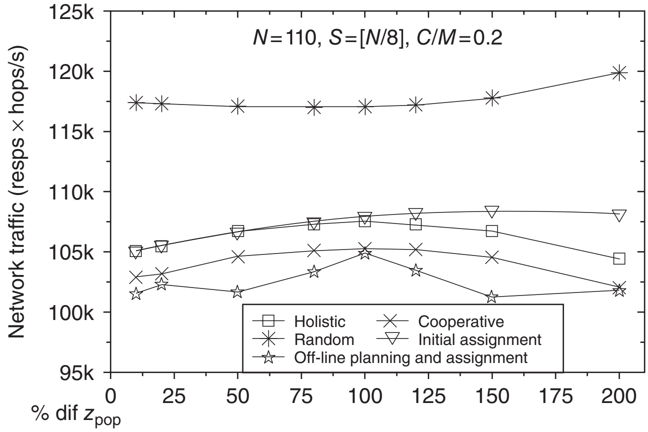

Figure 21.4 Performance of the online replacement algorithms.

Particularly, we initially assume that the popularities assigned to the nodes of the network are given by the vector $Z = (-1, -0.7, -0.5, 0, 0.5, 0.7, 1)$, and at each different experiment (different points in Figure 21.4) this vector changes by a given factor. This factor ranges from 10% to 200%. A change of 10% means that the new vector of popularities is $Z = (-0.9, -0.63, -0.45, 0, 0.45, 0.63, 0.9)$, whereas a change of 100% transforms the vector of popularities to $Z = (0, 0, 0, 0, 0, 0, 0)$, and a change of 200% inverts the vector. We also depict the performance of the initial cache assignment resulting from the initial planning with the new demand pattern as well as the performance of a totally new planning execution.

In Figure 21.4, we observe that the cooperative algorithm performs slightly better than the holistic, whereas the random assignment is at least 25% worse than the two presented algorithms. An interesting finding comes from the comparison of the presented online replacement algorithms with the performance of the initial cache assignment. We observe that when the changing factor of the initial popularities is smaller than 100%, the algorithms with the network-wide knowledge perform only $1 - 3\%$ better than the initial assignment; only when the changing factor is larger than 100% and the popularity vector reverts its sign, we observe a difference is the performance up to 10% regarding the overall network traffic. Finally, the two presented algorithms are performing close to the new planning and offline assignment, which means that a new planning is not needed very often and is useful only under extreme changes in the demand pattern.

Note that Figure 21.4 may also serve as a benchmark for the replacement components in their decision to reassign or not the stored items upon the detection of a change in the popularity pattern. Particularly, the difference between the network traffic cost of the initial storage assignment and the traffic cost after the completion of the algorithms combined with the communication complexity enables the replacement components to perform or skip the content reassignment. For a more detailed evaluation of the online replacement phase, refer to References 5, 6 where our numerical results provide evidence that network-wide knowledge and cooperation give significant performance benefits and reduce the time to convergence at the cost of additional message exchanges and computational effort. In particular, the cooperative algorithm provides the best performance regarding overall network traffic, but requires a high level of cooperation among the managers and hence is of very high computational and communication complexity. On the other hand, the holistic algorithm performs close to the cooperative but converges in a fraction of the iterations required by the cooperative. Thus, the comparison of the presented algorithms may serve as a valuable tool for the network manager so as to select the most appropriate algorithm for his needs, depending on specific network parameters (e.g., network size, number of information items, and volatility of the request pattern).

21.5 FUTURE RESEARCH DIRECTIONS

The work presented in this chapter can be extended in many ways such as optimizing different objectives to serve different quality of service (QoS) metrics and service-level agreements (SLAs) among the storage providers and the content providers. Also it would be interesting, as future work, to explore enhancements to the presented online replacement algorithms that would also take into consideration the cost of replacing the items at the replication points of the network, as well as the processing load of each replacement component when assigning items to them.

Another interesting extension of the presented work would be the identification and analysis of user and content mobility patterns. Considerable research has been dedicated to understand and model the behavior of mobile users from the social and technological perspectives. Most existing work on predicting mobility patterns, attachment points, and connectivity durations aims to minimize periods of disconnection. With ICN, however, content can "follow" mobile users, thus achieving shorter transaction periods; therefore, this work needs to be reevaluated. The outcome will be monitoring mechanisms, as well as algorithms for predicting the future connectivity points in order to move or migrate content accordingly. The presented replication framework could be exploited to identify strategic replication points at the edges of the network. The identification of these points should be based on (i) route selection and transmission scheduling and (ii) trade-offs between energy, delay, and cost. For example, replicating content closer to the user reduces energy consumption and delivery delay, as seen from the end-users point of view, as well as core-network traffic, as seen from the operator's point of view, but it increases replica deployment costs for the operator. Based on the investigation of such trade-offs, the outcome could be (i) the identification of strategic replication points close to the user, (ii) replication strategies to increase the amount of time content stays in the replication points, and (iii) the corresponding effect to the route selection and transmission scheduling algorithms.

21.6 CONCLUSION

In this chapter, we presented a generic three-phase framework as a contribution to the problem of information replication in ICN. The presented phases apply at different timescales and manage the content and the location of each replication point in the network targeting a specific objective. Moreover, in the newly presented online replacement phase, we presented two algorithms that differ in the amount of information that needs to be communicated and the required level of coordination among the replacement components. The online replacement decisions are based on real-time information, such as the observed popularity of the requests, and not on static offline predictions. Our numerical results provide evidence that network-wide knowledge and cooperation give significant performance benefits and reduce the time to convergence at the cost of additional message exchanges.

ACKNOWLEDGMENTS

V. Sourlas’ work is cofinanced by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II—investing in knowledge society through the European Social Fund.

REFERENCES

1. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs N, Braynard R. Networking named content. ACM CoNEXT; Rome, Italy; December 2009.
2. PURSUIT project. 2011. Available at http://www.fp7-pursuit.eu. Accessed 10 July 2013.
3. Li G, Cheung A, Hou S, Hu S, Muthusamy V, Sherafat R, Wun A, Jacobsen H, Manovski S. Historic data access in publish/ subscribe. Proceedings of DEBS; Toronto, Canada; 2007. p 80–84.
4. Sourlas V, Flegkas P, Paschos GS, Katsaros D, Tassiulas L. Storage planning and replica assignment in content-centric publish/subscribe networks. In S.I. on Internet-based Content Delivery. Comput Netw 2011;55(18):4021–4032.
5. Sourlas V, Flegkas P, Gkatzikis L, Tassiulas L. Autonomic cache management in information-centric networks. 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012); Hawaii, USA; April 2012. p 121–129.
6. Sourlas V, Gkatzikis L, Flegkas P, Tassiulas L. Distributed cache management in information-centric networks IEEE Transactions on Network and Service Management. September 2013;10(3):286–299.
7. Li B, Golin MJ, Ialiano GF, Deng X. On the optimal placement of web proxies in the Internet. Proceedings of INFOCOM; March 1999.
8. Cidon I, Kutten S, Soffer R. Optimal allocation of electronic content. Proceedings of INFOCOM; Anchorage; April 2001.
9. Kangasharju J, Roberts J, Ross K. Object replication strategies in content distribution networks. Comput Commun 2002;25:376–383.
10. Qiu L, Padmanabhan VN, Voelker G. On the placement of web server replicas. Proceedings of IEEE INFOCOM; Anchorage, USA; April 2001.

11. Arya V, Garg N, Khandekar R, Meyerson A, Munagala K, and Pandit V. Local search heuristics for k-median and facility location problems. Proceedings of 33rd ACM Symposium on Theory of Computing; 2001.
12. Charikar M, Guha S. Improved combinatorial algorithms for facility location and k-median problems. Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science; 1999. p 378–388.
13. Charikar M, Khuller S, Mount D, Narasimhan G. Facility location with outliers. Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms; Washington DC; January 2001.
14. Shmoys DB, Tardos E, Aardal KI. Approximation algorithms for facility location problems. Proceedings of the 29th Annual ACM Symposium on Theory of Computing; 1997. p 265–274.
15. E. Cronin, S. Jamin, C. Jin, T. Kurc, D. Raz and Y. Shavitt, Constrained mirror placement on the Internet, IEEE JSAC, 36(2), 2002.
16. Laoutaris N, Telelis O, Zissimopoulos V, Stavrakakis I. Distributed selfish replication. IEEE Trans Parallel Distrib Syst 2006;17(12):1401–1413.
17. Zaman S, Grosu D. A distributed algorithm for the replica placement problem. IEEE Trans Parallel and Distrib Syst 2011;22(9).
18. Borst S, Gupta V, Walid A. Distributed caching algorithms for content distribution networks. IEEE INOFCOM; San Diego, USA; March 2010.
19. Karlsson M, Karamanolis Ch, Mahalingam M. 2002. A framework for evaluating replica placement algorithms. Available at http://www.hpl.hp.com/techreports/2002/HPL-2002-21. Accessed 10 July 2013.
20. Jiang W, Zhang-Shen R, Rexford J, Chiang M. Cooperative content distribution and traffic engineering in an isp network. Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, ser. SIGMETRICS ’09; 2009. p 239–250.
21. Frank B, Poese I, Smaragdakis G, Uhlig S, Feldmann A. Content-aware traffic engineering. Proceedings of the ACM SIGMET-RICS/PERFORMANCE 2012 Joint International Conference on Measurement and Modeling of Computer Systems; 2012. p 413–414.
22. Kamiyama N, Mori T, Kawahara R, Harada S, Hasegawa H. ISP-operated CDN. Proceedings of the 28th IEEE International Conference on Computer Communications Workshops, ser. INFOCOM’09; 2009. p 49–54.
23. Cho K, Jung H, Lee M, Ko D, Kwon T, Choi Y. How can an ISP merge with a CDN? IEEE Commun Mag 2011;49(10):156–162.
24. Sourlas V, Flegkas P, Paschos GS, Katsaros D, Tassiulas L. Storing and replication in topic-based publish/subscribe networks. IEEE Globecom 2010 Next-Generation Networking and Internet Symposium; Miami, USA; December 2010.
25. Cohen R, Katzir L, Raz D. An efficient approximation for the generalized assignment problem. Inform Process Lett 2006;100:162–166.
26. Breslau L, Cao P, Fan L, Phillips G, Shenker S. Web caching and Zipf-like distributions: Evidence and implications. IEEE INFOCOM, NY; March 1999.
27. Padmanabhan VN, Qiu L. The content and access dynamics of a busy wed site; Stockholm, Sweden: ACM SIGCOMM; August 2000.
28. Newman MEJ. Power laws, pareto distributions and Zipfs law. Contemp Phys 2005;46:323–351.

29. Adamic LA, Huberman BA. Zipfs law and the Internet. Glottometrics 2002;3:143–150.
30. Knight S, Nguyen HX, Falkner N, Bowden R, Roughan M. The Internet topology zoo. IEEE JSAC 2011;29(9).
31. Flegkas P, Sourlas V, Parisis G, Trossen D. Storage replication in information-centric networking. International Conference on Computing, Networking and Communications (ICNC 2013); San Diego, USA; 2013. p 850–855.