A Value-Based Cache Replacement Approach for Information-Centric Networks

Fadi M. Al-Turjman[1], Ashraf E. Al-Fagih[2], and Hossam S. Hassanein[3]
[1]School of Engineering
University of Guelph, Guelph, Ontario, N1G 2W1, Canada
fadi@uoguelph.ca
[2]Information and Computer Science Department
King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
alfagih@kfupm.edu.sa
[3]School of Computing
Queen's University, Kingston, Ontario, K7L 3N6, Canada
hossam@cs.queensu.ca

***Abstract*— Information-Centric Networks (ICNs) represent a content-based model which addresses user's requests regardless of the content's location or the nature of its original publisher. The performance of an ICN is highly dependent on replicating the content across the caches of a multitude of nodes in the network. Given the high data turnover rates of contemporary applications and the finite nature of caching space, efficient caching algorithms play a crucial role in determining which data item can be safely dropped in order to accommodate for more important items. In this paper, we present a value-based cache replacement approach that executes a Least Valuable First (LVF) policy. Our approach employs a utility function that uses delay, popularity and age parameters to determine which item to drop from the cache. We present simulation results comparing our approach to other dominant cache replacement policies under varying conditions such as data popularity, in-network cache ratio and connectivity degree. Results show that our approach outperforms in terms of time-to-hit, hit rate, in-network delay and data publisher load.**

Keywords—information-centric networks; caching; disruption-tolerance; data clouds.

I. INTRODUCTION

Information-Centric Networking (ICN) provides an architectural model that primarily adheres to the user's content requests regardless of the nature or locale of the content's hosts. The traditional Internet model is suitable for nodes that intentionally seek communication between each other. However, the present networking paradigm suffers from significant inefficiencies with its use of physical networking resources. ICNs promise to eliminate significant communication overhead associated with the distribution of commonly accessed data files. This is done by adopting an architecture that is based on Named Data Objects (NDOs) such as web pages, videos, or other forms of content [1]. We note that the terms *NDO*, *content* and *data items* will be used interchangeably across this paper.

According to the ICN model, entities include nodes, caches attached to nodes and the data items [2]. A node can take on three roles. It is either a publisher, a user, or an intermediary node. A publisher is defined as an entity that is able to serve a particular data item. Communication in ICNs is driven by users requesting NDOs and named publishers making NDOs available to users. User nodes are defined as those placing requests for data items. Users solely play a consumer role and publishers solely play a producer role. Intermediary nodes facilitate the communication between users and publishers by providing routing and buffering resources.

One of the prominent features of the ICN concept is that content caches can be attached to nodes [3]. The ICN design takes into account that data is often only published once and is copied to different locations many times. This is handled using caches at all levels of the network. Caches reduce the amount of traffic going through the network by fulfilling packet requests away from the bottleneck created by publishing information at one location. Not only does this method shorten the response time for multiple data requests to the same server (publisher), it also reduces the distance requests must travel if data is corrupted along the way from the producer to the consumer [4], [5].

In ICN designs, when a request for content is received, the supplying node either: i) responds with the content directly if it has the data cached, or ii) requests the content from its neighboring suppliers if it does not have the content cached. The latter option implies caching the content when this request is addressed. However, since caching space is a finite resource (represented by storage spaces in routers, sensors, smart devices, etc.), one of the major challenges of the ICN concept is related to the caching replacement policy. That is to say, which data item to drop if there is no free space in the cache?

Traditional cache replacement policies [6] were optimized for computer architectures or databases, which have distinct characteristics from networking. Selecting the most efficient ICN cache replacement approach has implications on the performance of the network as a whole in terms of NDO hit ratio, time-to-hit, in-network delay and the publishers' load. Each of these metrics may be considered

978-1-4799-0540-9/13/$31.00 ©2013 IEEE

separately as a design objective for ICN caching policy and depending on the application in hand. However, since a single ICN layout may serve a multitude of applications/users with varying data requests, we argue that defining a dynamic utility function that sets a *value* for the data depending on users' requesting trends represents the best approach to manage ICN caches.

To this end, we summarize our contributions as follows:

- We provide a novel utility function that sets a value to each data item (content) of an ICN based on user-specified utility metrics.
- We provide a cache replacement algorithm called Least Value First (LVF) that is based on the value of the data item as oppose to its sole age, location with respect of the queue or frequency of usage, for instance.
- We compare our LVF approach with two dominant cache replacement approaches: First In First Out (FIFO) and Least Recently Used (LRU) with regard to various performance metrics under a variety of parameters including cache percentage in the network, time-to-hit and connectivity degree.

The remainder of this paper is organized as follows: Section II provides background on caching approaches in LCNs. Section III introduces our system models and problem description. Section IV explains the LVF utility function and algorithm. Section V evaluates the performance results of LVF in comparison to other caching policies. Finally, Section VI concludes this work.

II. BACKGROUND

Many studies introduce the ICN paradigm as a means to progress the Internet away from its current reliance on purely point-to-point primitives and, to this end, have proposed detailed designs that make the Internet more data-oriented or content-centric [7]-[10]. This idea is hardly new; the TRIAD project [7] described an ICN like design. In addition, the IETF in [8] reaffirmed the point that we should move towards a world where "the primitive operation in displaying a web page is no longer an end-to-end connection to the web server, but the delivery of a named block of data"[8].

As for the caching approaches in ICNs, there tends to be a distinction between caching at the network edge, as in peer-to-peer (P2P) and other overlay networks, and in-network caching (e.g. transparent web caches). The authors of [4] proposed a scheme to reduce both network delay and publisher load by providing lightweight cooperative mechanism under age-based control. Age is dynamically defined here by each router while assuming that older packets will be closer to the edge of the network.

A replica is replaced by other object(s) if both of the following conditions are satisfied: i) The age of the content has expired and ii) The cache memory of the node has been full. This scheme, however, correlates the concept of age with the location of the packet regardless of the users' query trends and does not address to other requesting metrics.

In [10], an ICN architecture is realized as an overlay network, based on the deployment of additional infrastructure inside access networks. Namely, ISPs deploy and control *Overlay Access Routers* (OARs). These OARs establish an overlay routing substrate for the forwarding of data based on information identifiers. Needless to say that the assumption of additional components adds to the overall cost of this solution.

The approach introduced in [11] bridges ICNs and Redundancy Elimination (RE). RE is a popular technique to detect and eliminate similar byte streams from network traffic irrespective of traffic source or protocol. Such systems require that the two end points are able to cache recent packets and have their caches synchronized. Redundancy, however, may be required particularly with highly popular content.

The authors of [12] propose alternative strategies, based on off-path caching in order to avoid redundant caching of data items and a more efficient use of caching space. Based on a mathematical model for LRU cache management, the authors of [12] analyze their strategies with respect to hit efficiency, delay, and power consumption and provide a competitive evaluation to assess the potential increase of energy efficiency with off-path caching. However, the notion of delay here is related to the hop-count rather than the delay-sensitivity of the data, which is a different metric.

Our approach divides the load between all the main components of an ICN architecture, including the user [13], in order to provide a user-assisted cache replacement policy that employs user input to decide on the value of the content according to delay, age and popularity metrics without requiring additional infrastructures or complicated formulations as explained in the following sections.

III. SYSTEM MODELS & PROBLEM STATEMENT

In this section, we present our ICN network model in addition to its corresponding delay, popularity and age models. We then present our problem statement.

A. ICN Network Model

The components comprising our LCN model are as follows:

1) **Publishers** that represent nodes capable of publishing NDOs into the ICN. According to our model, publishers may include sensors, RFID tags, tablet PCs, smart phones and any ambient nodes that may generate NDO content in any form.
2) **Intermediate Nodes** (INs) are in-network devices that perform the processing and caching of NDOs such as routers, switches, relays, etc. However, INs cannot publish data by themselves.
3) **Caches** that are part of all nodes. We particularly focus here on IN caches since they hold the important role of facilitating in-network delay in addition to increasing the NDO hit-rate by efficiently addressing the high demand on popular data items.
4) **Users** (consumers) are the NDO requesters and receivers

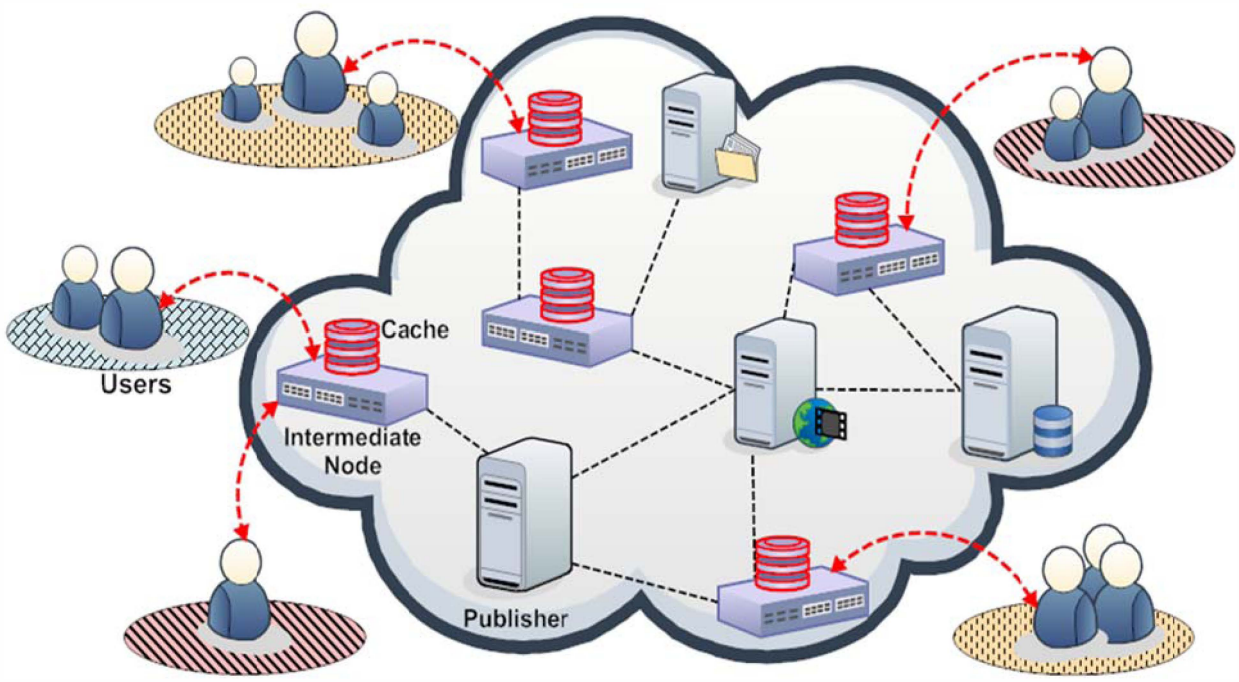

Figure 1. Network model of an ICN

in the ICN. They are represented in reality by hand-held smart phones, laptops, smart vehicles, etc. Based on geographic proximity; a user may be viewed by an IN as a member of group. The nature of group requests set the parameters of our caching policy at the INs they connect to by collectively providing the requesting trends.

Fig. 1 depicts the aforementioned ICN components.

B. Delay Model

In-network delays, disruptions and disconnections in ICNs may occur when a popular NDO source links to a smaller site, causing an increase in traffic in an effect known as *slashdotting*. End-to-end communication between users and source nodes is often difficult to achieve in challenged networks due to sparse connectivity[14], disruption [15] and node mobility [16]. The dense network topologies associated with ICNs result in a long multi-hop path consisting of heterogeneous devices. In addition, these devices might not have an immediate communication link with the next hop which causes extended delay periods while holding data until the communication link is available; either because it is a mobile node with a limited speed s, or because it is a static node buffering data until a mobile collector arrives and picks its data. Thus, the delay/disruption components we have to consider are dominated by the queuing/propagation delay ψ, and the holding delay ϵ. These components are extremely dependent on the relaying hops count and the node mobility patterns. Accordingly,

$$\psi \propto n\,, \tag{1}$$

where n is the total number of hops between the content's packet source and the consumer. And the holding delay ϵ is determined based on the following:

$$\epsilon \propto \left(\frac{d}{s} + T_p\right), \tag{2}$$

where d is the distance separating a mobile data holder or collector from the next hop on path towards the data consumer, and T_p is the pause time this mobile node would experience during its physical motion. Hence, we define a discretized disruption step D_i, which is the delay a packet I would experience until it reaches the data consumer, as follows

$$D_i = \lceil \psi + \epsilon \rceil \tag{3}$$

Thus, we define a normalized D'_i as

$$D'_i = \frac{D_i}{D_{max}} \tag{4}$$

where D_{max} is the maximum expected delay.

C. Popularity Model

Popularity is represented by the frequency of which a certain NDO is being requested by users. The higher the popularity of an item the least probable it is going to be removed from the cache.

In a given geographical vicinity G and within a given operational cycle T, we define the popularity of a given NDO_i at any IN by

$$Pop_{NDO_i} = \frac{Req_{NDO_i}}{Req_{Total}}, \tag{5}$$

where Req_{NDO_i} is the total number of requests for NDO_i received by the IN and Req_{Total} is the total number of NDO requests received by that same IN within a particular operational cycle.

D. Age Model

Our age model resembles the FIFO caching policy in the sense that content is dropped from the queue if both of the following conditions are satisfied: i) the age of the content has expired. ii) The node's cache is full. An NDO's age is represented by its time-to-live (TTL) which is assigned either by the publisher or updated by an IN according to application.

According to our approach, the probability of dropping a packet from the cache may be proportional (either directly or inversely) to its age. That is to say, newer packets may be sacrificed if the users' requesting trends prefer older packets. The alternative case also holds. Hence,

$$Drop_{NDO_i} \propto TTL_{NDO_i} \quad (6)$$

Based on the above formulation, the targeted LVF approach aims towards the following:

Given an ICN network with N nodes and varying percentage of caching capacities and connectivity degrees, find the most appropriate caching approach such that the least valuable data item is first dropped.

IV. Least Valuable First Approach

Our approach is based on the aforementioned system models to achieve an LVF cache management policy to handle the following three types of content:

1) Delay-based content: Delay is measured by in-network delay which consists of several components such as processing delay, transmission delay and caching delay as we aforementioned in section III. The delay sensitivity of NDOs is a measure specified by the requesting user to indicate how long the consumer is willing to wait for the NDO. Examples of delay-sensitive NDOs are related to applications serving areas of emergencies or catastrophe (e.g. request of information on a certain health condition or location of nearest fire station).

2) Demand-based content: This is a measure of NDO's popularity which is specified by the frequency of requesting each NDO.

3) Age-based content: The age of a content item is measured by its TTL count. Age applies to any content item regardless of its nature or origin. However, some NDOs are more sensitive to aging in a manner that renders them *expired* if their age exceeds a specific time limit. For instance, if a user requests information on the weather within the next 24 hours, or the traffic updates for the coming 30 minutes, then any related NDO that does not cover these time intervals is useless.

Accordingly, LVF cache management approach utilizes three parameters to set a *value* ($Value_{NDO}$) per NDO_i. This value is dependent on the history of the user's requests received by INs within each operational cycle. At the beginning of each cycle, every IN resets the value of its cached NDOs according to the following function:

$$Value_{NDO_i} = \alpha \,.\, D'_i + \beta \,.\, Pop_{NDO_i} + \gamma . Drop_{NDO_i}, \quad (7)$$

where $\alpha, \beta, and\ \gamma$ are tuning parameters specified based on the user group at requesting time of the NDO.

The advantage of the LVF is in its ability to allow for priority customization based on the user-provided parameters and how each parameter affects data flow through the network. Therefore, to improve the basic priority caching method, the weights of each of the parameters can be adjusted to find the most efficient approach. For example, the rank of popularity can be multiplied by a factor of two to see if the popularity is the most important metric when it comes to the priority caching.

In terms of advantages over the other two caching methods (i.e. FIFO and LRU), LVF takes into account more than one metric to determine which packets will stay in the cache. This allows for a much better prediction on which packets may be used in the future. The delay sensitivity parameter serves in insuring minimal delay while delivering requested packet. The popularity parameter is important as it takes into account the most used packets and if that trend is followed, the odds are that these packets will be used the most in the future. The packet age value is also valuable since it takes into account packets in the cache that have not been used in a long period of time and removes them in favor of more relevant data.

Fig. 2 depicts a use case in which users are divided into three separate groups based on their requesting preferences. The users in group *A* put more emphases on delay-based NDO, users in group *B* are more interested in a popular NDO (e.g. a specific video clip) while users in group *C* are mostly requesting some age-based NDO. Corresponding INs in each area reply accordingly by assigning higher weights to delay, popularity and age parameters in Eq. (7).

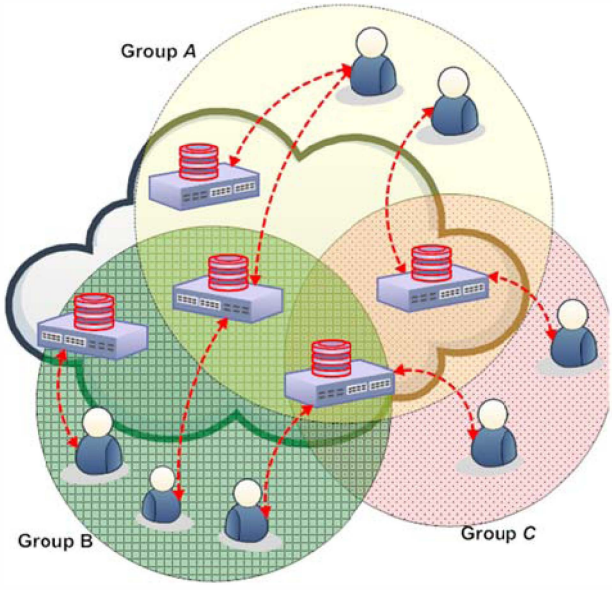

Figure 2. Use case of caching parameters calculation.

INs within the intersection areas (of Fig. 2) perform more complicated weight calculations according to the corresponding requests of users connecting to them. Algorithm 1 provides the steps to be executed by each IN if its cache is full to drop the least valuable content item.

V. PERFORMANCE EVALUATION

In this section, we compare our proposed LVF caching approach against two dominant cache management approaches; FIFO and LRU. Prior to that, we introduce the metrics and parameters we based our performance evaluation on.

Algorithm 1: Drop least valuable item from cache.

```
1.  Function LVF (content)
2.  Input
3.  content: A data item within the ICN.
4.  Begin
5.  for each IN, do
6.    for each duty cycle, do
7.      Set value of each content item in the cache   // Eq. (7)
8.      if cache_full
9.        Check history of user requests
10.       Drop least valuable content item
11.     End if
12.   End for
13. End for
14. End
```

A. Performance Metrics & Parameters

To compare the performance of the proposed LVF approach, we track four metrics to be able to make quantitative conclusions in line with the objectives described in our Introduction. First, we note that statistics will not be collected until all caches have been filled. Until this occurs, we will not start simulating cache failures. We simulate long-term performance of the network with low computational overhead so that we may run more simulation instances with our limited resources. If we start collecting statistics before the caches are filled, the simulation runtime would have to be longer to drown out the penalties of the initial cache misses which become irrelevant in long-term performance. The four routing metrics of interest are as follows:

1) Cache hitting ratio

Cache hitting ratio for a single cache is simply the fraction of time a request arrives at a node to which that cache is attached but does not contain the requested data item. Average cache hitting ratio is the average hitting ratio over all caches, weighting each cache by the number of requests that pass through it. We will track publisher load by keeping track of the total fraction of requests that had to be satisfied by a publisher node. Later, we average over all the publisher loads to produce a single figure for the inner simulation. To determine which cache policies are effective, we prefer to look at average time to hit data and hitting ratio more than publisher load, but we generally expect publisher load to improve as the other metrics improve as well. In other words, we hypothesize that our ICN efficacy metrics are also representative of caching efficacy, and that better caching policies mean a better ICN in general.

2) Time-to-hit-data

To find time-to-hit (TTH), we simply log all the total costs of the request and response paths incurred by every consumer node over all request and receive transactions. We are not modeling varying content size to simplify the simulation, so it suffices to log just the path costs. From a collective perspective, a successful ICN will minimize the total average TTH per request. However, publisher load and hitting ratio are also important metrics that give complementary indications of the efficacy of the ICN.

3) In-network delay

This metric was explained earlier as the result of disruption or disconnections across the end-to-end paths. However, the more efficient the caching policy, the higher the probability a replica of the requested content is available in a nearby IN, which consequently reduces delay. Note that we differentiate between time-to-hit-data and in-network delay since the two metrics may differ due to mobility or disruption conditions.

4) Average request per publisher (ARP)

This metric is measured in request per hour (*req/hr*) and it represents the average load per publisher in an ICN network. Due to in-network caching, in addition to the selected cache replacement policy which participate in disseminating/distributing the in-network data, the publisher's load can be reduced. Thus, bandwidth and computing resources of the publisher can be saved.

Meanwhile, many of the candidate parameters must remain fixed through all simulation instances, and hence are actually constants. In particular, the variables of our approach (and those used for experiment design), are as follow:

1) Percintage of nodes with caches (PoC)

This parameter is utilized for controlling the extent of caching in an ICN. In particular, it allows controlling how many nodes in a network instance are attached to a cache. The more nodes that have caches attached the greater extent of caching we will have in an ICN. By varying this parameter, we can study the sensitivity of metrics like time-to-hit-data to the caching extent. This will allow us to answer questions like, how many caches are needed to feel the benefits of caching in ICNs? Obviously, if every node needs to have a very large cache, we lose the benefits, especially since many mobile devices cannot afford caches. The ideal outcome is that we find major improvements in time-to-hit-data with medial increases in the proportion of nodes with caches. During topology initialization of a simulation instance, this proportion of nodes is chosen to be attached to caches. Nodes without caches effectively have cache size equal to zero.

2) Connectivity degree

During topology generation, when determining whether any two nodes are connected, we use a uniform random variable $(RV) \sim U[0,1]$ to make the connectivity matrix, and then use a "connectivity threshold" to determine which of these matrix elements represent a connection and which do not. Being above this threshold means that the node is connected. The consumers with variable location experience varying connectivity during simulation runtime, and this connectivity threshold is again used to determine their connectivity throughout the simulation. Connectivity degree is calculated as

$$Conn.Degree = N \times (1 - Conn.\ Threshold) \qquad (8)$$

We want to study the effect of increasing connectivity degree to quantify the resulting marginal benefit in TTH.

3) Data pobularity

It represents how frequent a specific piece of data is requested. This metric is measured in percentage with respect to other requested data in our simulation. Our objective is to determine the extent to which the frequency of popularity turnover thwarts the efficacy of ICNs. Consider the case where content items are rapidly appearing and disappearing from the universe of possible content items. We wish to confirm that high content turnover will not have a detrimental effect on our caching approach efficacy in ICNs.

B. Simulation Setup

Our simulation is initialized by generating a random graph topology of N nodes. 10% are publishers, 60% are INs and 30% are users. We then produce a symmetrical connectivity matrix. Each element in the matrix takes on an outcome of a uniform random variable $(RV) \sim U[0,1]$. A simulation parameter gives the threshold above which we assume connectivity. Once connectivity is determined, the edge between $N1$ and $N2$ (where $N1$ and $N2$ are any connected nodes in the network) will be assigned a random value $\sim U[a,b]$, where $[a,b]$ is an input, representing the cost (in terms of time) of transferring data from $N1$ to $N2$ and vice versa. That is, we assume all channels are bidirectional. During the simulation, the graph will only be changed by relocating a fraction of the consumers (representing mobile nodes). Of the consumer nodes, we randomly pick a fraction F, given as a simulation parameter, which will experience variable connectivity. The goal here is to simulate mobile consumers, such as those we find in vehicular networks with s and T_p following a uniform random distribution function. It is very important to keep in mind that we are assuming the time it takes for a data request to be satisfied is considered negligible compared to the time between which the topology experiences changes.

A proportion of the nodes randomly determined will be attached to caches. This proportion is specified as one of the input parameters of the simulation and is a crucial input whose effect we are monitoring to determine the efficacy of caching in an ICN. The cache sizes will vary by the node, taking on the outcome of a discrete RV with uniform Popularity Mass Function (PMF) ranging from 1 to N.

Data items can be replaced during the simulation, and a measure of how commonly each item is accessed is recorded. A reasonable assumption for a successfully implemented ICN is that every item only needs, and only has, one server source from which it is originally disseminated. That is, each data item only has one publisher node. During simulation initialization, the data items will be fully populated.

C. Simulation Results

Figs. 3-8 represent our simulation results. Our first and foremost objective was to confirm that increasing the extent of caching in ICNs will reduce TTH for all cache policies. This is exactly confirmed in the following figures.

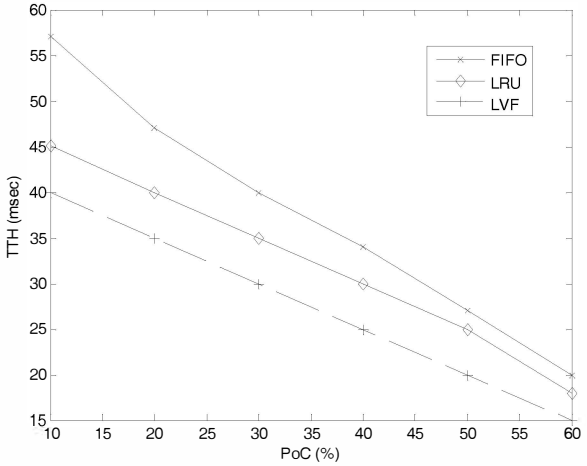

Figure 3. Time to hit ratio vs. percentage of nodes with caches (with connectivity degree = 30).

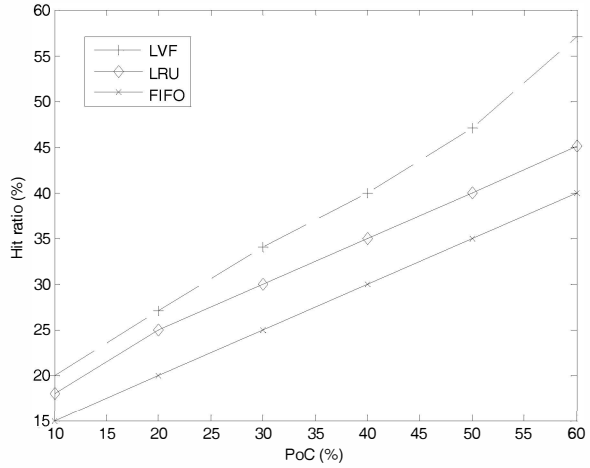

Figure 4. Hit ratio vs. percentage of nodes with caches (with connectivity degree = 30).

In Figs. 3 and 4, the parameter is the proportion of nodes in the ICN that are attached to caches (PoC). The extent of cache availability increases proportionally. We see in Fig. 3 that TTH, our primary performance metric, is reduced for all cache policies. Nevertheless, the LVF caching policy performs best at higher proportions of nodes attached to caches.

Fig. 4 shows a monotonically increase in the hit ratio for the three approaches. We remark that the LVF outperforms the other two approaches due to its ability on replacing the most appropriate data.

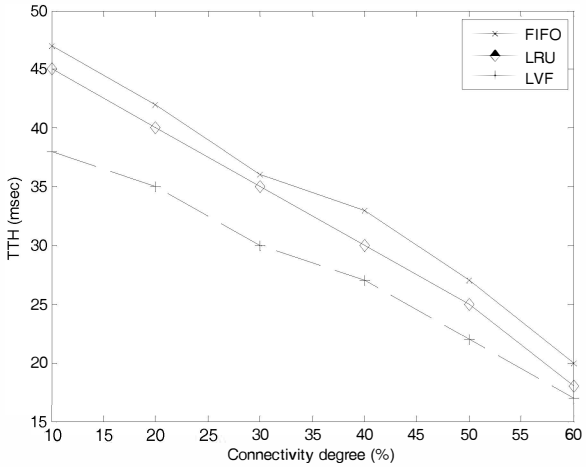

Figure 5. Time to hit ratio vs. the connectivity degree percentage.

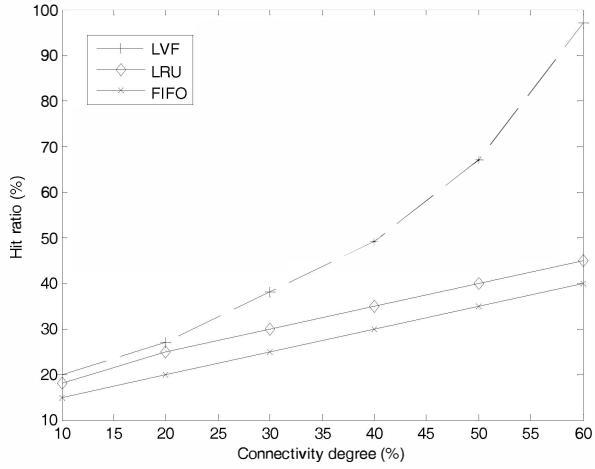

Figure 6. Hit ratio vs. the connectivity degree percentage.

For the following figures (Figs. 5-7), connectivity degree, is the testified parameter. Fig. 5 shows a monotonically decrease in time-to-hit (TTH) data metric as the network connectivity enhances with all approaches. However, our LVF approach outperforms the other two approaches as it is less dependent on the network itself. LVF is more dependent on data type which is a very desirable feature in an ICN network. We can expect that increasing connectivity will have a major effect on reducing the time to meet data. This is because by increasing the connectivity, we are effectively increasing the number of routing channels available in the system, giving rise to shortest routing paths.

Fig. 6 shows the hit ratio performance while applying varying connectivity degrees. This figure shows a noticeable trend while applying our LVF approach. The hit ratio increases exponentially as the network connectivity increase, unlike the other two approaches where the hit ratio increases linearly.

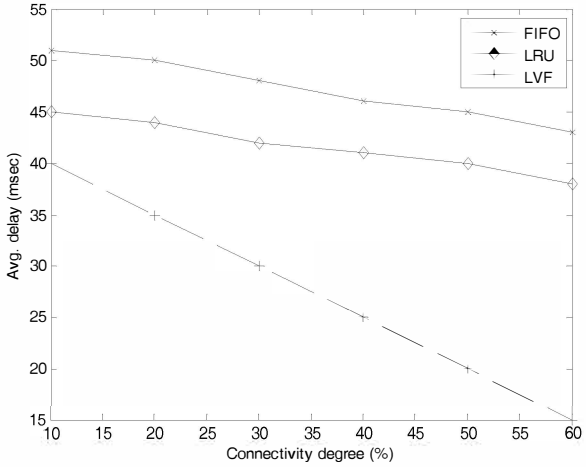

Figure 7. Avg. in network delay vs. the connectivity degree percentage.

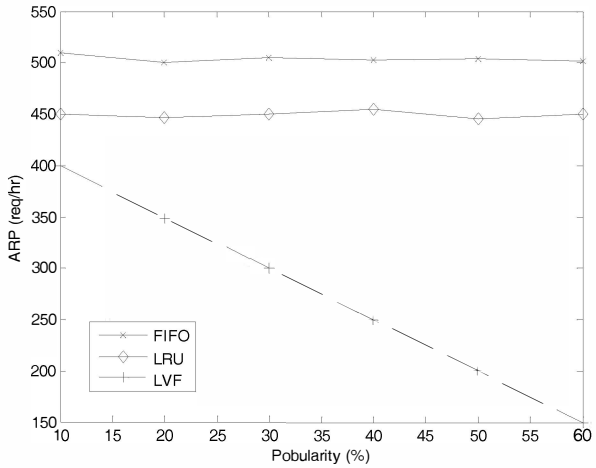

Figure 8. Publisher load vs. the data popularity.

Fig. 7 depicts that the best approach in terms of delay is the LVF approach. This can be returned to the consideration

of the delay factor in Eq. (7) while deciding which data to replace.

Finally, Fig. 8 shows the effect of data popularity on the three cache replacement approaches in terms of the data publisher load. Our LVF approach outperforms the FIFO and LRU approaches significantly as the requested data popularity increases. This is a very desirable feature in a data-centric network like the ICNs.

VI. CONCLUSION

In this paper we introduce LVF – a Least Value First caching policy for ICN networks. Our approach caters for heterogeneous (mobile/static) data publishers and/or intermediate routing nodes. According to our approach, a value to each in-network data item (content) is set based on user-specified utility metrics as opposed to its sole age, location in the queue, or its usage frequency. LVF realizes delay-tolerant data requirements. It provides a dynamic data replacement policy that is based on the consumers' requesting trends. Thus, LVF maximizes the consumer's gain according to delay limit, data popularity, and time-to-live metrics while determining the best data item to be dropped from filled caches. In addition, LVF maximizes the gain of the publishers by reducing their load. We provide simulation results showing the efficiency of LVF when compared to FIFO and LRU caching policies. Our results show that the LVF exhibits superior performance in terms of publisher load, end-to-end delays time-to-hit data, and hit ratio.

VII. ACKNOWLEDGEMENT

This research is partially funded by King Fahd University of Petroleum and Minerals, and a grant from the Ontario Ministry of Economic Development and Innovation under the Ontario Research Fund-Research Excellence (ORF-RE) program.

REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and Börje Ohlman. "A survey of information-centric networking," IEEE Communications Magazine, vol. 50 , no. 7, pp. 26-36, 2012.

[2] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," In Proceedings of the 10th ACM Workshop on Hot Topics in Networks, Cambridge, MA, Nov. 2011, pp. 1-6.

[3] S. Arianfar, P. Nikander, and Jörg Ott, "Packet-level caching for information-centric networking," Finnish ICT-SHOK Future Internet Project, Technical Report, 2010.

[4] Z. Ming, M. Xu, and D. Wang "Age-based cooperative caching in information-centric networks," In IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, Canad, Apr. 27 – May 2, 2012, pp. 268-273.

[5] A. Dan and D. Towsley. An approximate analysis of the LRU and FIFO buffer replacement schemes. vol. 18. no. 1. ACM, 1990.

[6] A. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell. A Hierarchical Internet Object Cache. In Proc. of USENIX, 1996.

[7] M. Gritter and D. R. Cheriton. TRIAD: A New Next-Generation Internet Architecture. Stanford University, July 2000.

[8] B. Baccala. Data-oriented Networking. Internet draft, IETF, Aug. 2002.

[9] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," In NETWORKING, pp. 27-40. Springer Berlin Heidelberg, 2012.

[10] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache: An overlay architecture for information-centric networking," Computer Networks, vol. 55, no. 4, 2011, pp. 936-947.

[11] D. Perino, M. Varvello, and K. Puttaswamy, "ICN-RE: redundancy elimination for information-centric networking," In Proceedings of the 2nd Edition of the ICN Workshop on Information-Centric Networking, Helsinky, Finland, Aug. 2012, pp. 91-96.

[12] M. Draxler and H. Karl, "Efficiency of On-path and off-path caching strategies in information centric networks." In IEEE International Conference on Green Computing and Communications (GreenCom), Besançon, France, 20-23 Nov. 2012, pp. 581-587.

[13] H. Y. Lee and Akihiro Nakao, "Efficient user-assisted content distribution over information-centric network," NETWORKING, pp. 1-12. Springer Berlin Heidelberg, 2012.

[14] F. Al-Turjman, W. Alsalih, and H. Hassanein, "Towards augmented connectivity in federated wireless sensor networks", In Proc. of the IEEE International Conference on Wireless Communications and Networking (WCNC'12), Paris, France, Sep. 2012, pp. 1882 -1886.

[15] F. Al-Turjman, A. Al-Fagih, W. Alsalih, and H. Hassanein, "A delay-tolerant framework for integrated RSNs in IoT," Elsevier: Computer Communications Journal, vol. 36, no. 9, pp. 998–1010, May, 2013.

[16] F. Al-Turjman, H. Hassanein, and M. Ibnkahla, "Optimized relay repositioning for wireless sensor networks applied in environmental applications", In Proc. of the IEEE International Wireless Communications and Mobile Computing conf. (IWCMC11), Istanbul, Turkey, Jul. 2011, pp. 1860-1864.