Journal of Ambient Intelligence and Humanized Computing (2023) 14:6735–6749
https://doi.org/10.1007/s12652-021-03538-6

ORIGINAL RESEARCH

Optimization of content-centric networking based IoT systems using partitioning-heuristics in-network caching

Sumit Kumar[1] · Rajeev Tiwari[1]

Received: 16 November 2020 / Accepted: 7 October 2021 / Published online: 21 October 2021

Abstract
Internet of Things (IoT) has emerged as a novel paradigm that focuses on connecting a large number of devices with the Internet infrastructure. To address the performance requirements of IoT devices, the Content-Centric Networking (CCN) becomes an encouraging future Internet architecture that emphasizes name-based content access, instead of searching for the host-location in the network. The in-network content caching is an essential characteristic for rapid information dissemination and efficient content delivery in the CCN. To this end, a novel content caching scheme has been proposed for comprehensive utilization of the available caching resources. The proposed scheme partitions the CCN-enabled IoT networks hierarchically to reduce content redundancy and excessive cache replacement operations. For content caching decisions, the proposed caching strategy considers normalized distance-based metrics along with dynamic threshold heuristics to improve content retrieval delay by placing the contents near the IoT devices. Extensive simulation analysis on realistic network configurations demonstrates that the proposed caching scheme outperforms the existing competing content placement strategies on performance parameters such as network cache hit-ratio, hop-count, delay, and average network traffic. Thus, the proposed caching scheme becomes more promising for CCN based IoT applications.

Keywords Internet of Things · Content-centric networking · Content caching · Network partition · CCN performance

1 Introduction

The IoT is a rapidly growing environment, which is forming the future of the present world. The IoT-based connected world becomes a reality that contains various domains such as smart cities, smart wearables, healthcare, agriculture, Internet of vehicles, etc (Meddeb et al. 2017; Kumar et al. 2015). The IoT networks appeared to be a promising architecture that connects different types of objects to the Internet such as home appliances (washing machine, microwave, refrigerator), vehicles, tracking systems, automatic irrigation systems, etc. as mentioned in Arshad et al. (2018) and Mishra et al. (2018). In recent, the number of IoT devices associated with cyberspace is growing exponentially (Jindal et al. 2019). Therefore, the generation and handling of a large amount of data become evident on such networks that raises several issues on current IP-based Internet architecture (Atzori et al. 2010). The increase in the number of IoT devices requires an efficient addressing protocol that uniquely identifies each device in the network. The current Internet architecture, IPv4 has only 32 bits of addressing mechanism which is not sufficient for addressing IoT devices (Shang et al. 2016). Therefore, the majority of IoT systems relies on IPv6 protocol for addressing and routing purposes. However, in IPv6 protocol, the 50 bytes long header and minimum 1280 bytes payload size have introduced communication overhead and increase the complexity of the network stacks (Mars et al. 2019). Moreover, due to the host-centric approach of IP-based Internet architectures, the requester searches for the host to access the required content instead of searching the content directly in the network. This increases the load on the server, network traffic, and congestions during content retrieval. To reduce the server load and content retrieval delay, some significant caching schemes are proposed in wireless networks and vehicular ad-hoc networks such as Tiwari and Kumar (2015, 2016) for IP-based networks. However, these schemes also suffer

✉ Sumit Kumar
sumitaggarwal001@gmail.com

Rajeev Tiwari
rajeev.tiwari@ddn.upes.ac.in

[1] School of Computer Science, University of Petroleum and Energy studies, Dehradun, India

from the inherent restrictions of IP-based networks such as inefficient content retrieval, congestion control, and inter-domain routing protocols.

To address these issues, various Internet architectures have been proposed for the future of IoT systems. In this direction, CCN is one of the most recent advancements, where IoT devices retrieve the contents by content names and provide name-based routing instead of searching for the host in the network (Jacobson et al. 2007). CCN becomes the most widely accepted potential Internet architecture that improves Quality-of-Service (QoS) for its users in terms of lesser delay in accessing the requested content, network traffic, and scalability with reduced cost (Xylomenos et al. 2013).

The in-network caching is a decisive characteristic of CCN to improve QoS in IoT systems (Zhang et al. 2020). The scheme mentioned in Vural et al. (2014) suggests that the in-network content caching strategies reduce the network traffic and average delay by disseminating the content copies in the network and a requester can access data from the intermediate router instead of searching for the specific host. Therefore, the performance of an IoT system is highly reliant on the effectiveness of the in-network caching strategy to disseminate the contents in such a manner that the IoT devices would experience minimal delay in accessing the contents (Nour et al. 2019). An in-network caching strategy identifies content placement and content replacement heuristics, where content placement strategy takes decisions regarding the selection of routers on which the content should be placed and content replacement strategy selects the content for eviction from the cache once it is full.

The contribution of this work is towards in-network content caching in IoT systems. Precisely, this article investigates the existing caching schemes implemented in the CCN and analyse the effect of various parameters used in content caching strategies to examine network performance. Then, to improve hit-ratio, average network delay, and network bandwidth utilization, a novel content caching scheme has been proposed in this paper that divides the IoT network into several partitions (clusters Ma et al. 2020) and takes caching decisions based on the hop-count parameter. The proposed hierarchical network partitioning strategy is used to control the caching operations in the partitions. When the IoT device (requester) and the content provider (router) exist in the same partition, the intermediate on-path routers do not perform caching operations during content forwarding to reduce content retrieval delay, redundancy, and cache replacement operations in the network. The proposed heuristics also distributes the caching capacity fairly among different transmission paths in the partition. If the IoT device and the provider reside in different partitions, then each intermediate partition would cache at most one replica of the forwarded content based on the caching policy. During content forwarding, the proposed scheme takes content placement decisions based on partition configurations and distance-based metrics from the content provider. Using the proposed caching scheme, the content placement probability increases with increase in the distance traversed by the content. When the content is cached in the router of a partition, then the remaining routers of that partition would forward the content towards the requester without further caching operations. This mechanism reduces the computational delay in the intermediate routers. It has been argued that deploying this partitioning-based caching mechanism would improve content diversity in the network and thus, results in reduced content retrieval delay for the IoT devices. The summary of the contributions of this work is as follows:

- For the comprehensive utilization of available cache resources in IoT systems, a novel content caching scheme has been proposed based on the network partitioning and distance of on-path routers from the content provider parameters.
- The hierarchical network partitioning strategy has been implemented for network divisions and the hop-count parameter is considered for effective content placement decisions along with Least-Recently Used (LRU) replacement strategy.
- Extensive simulation analysis on standard IoT network is performed that verifies the performance gain of the proposed caching scheme over existing peer competing schemes on parameters such as network hit-ratio, average network hop-count, average delay, and network traffic.

In the coming sections, the paper is organized with an overview of the CCN system and its working in Sect. 2. Then, related works are covered in Sect. 3, which has a comparison table at the end of the section. The system model details are given in Sect. 4. The proposed hierarchical partitioning is designed in Sect. 5 and the proposed partitioning-based caching scheme is defined in Sect. 6. Then, the proposed scheme is evaluated for its performance with peer competing schemes in Sect. 7 and at last the conclusion is drawn based on the outcome of work in Sect. 8.

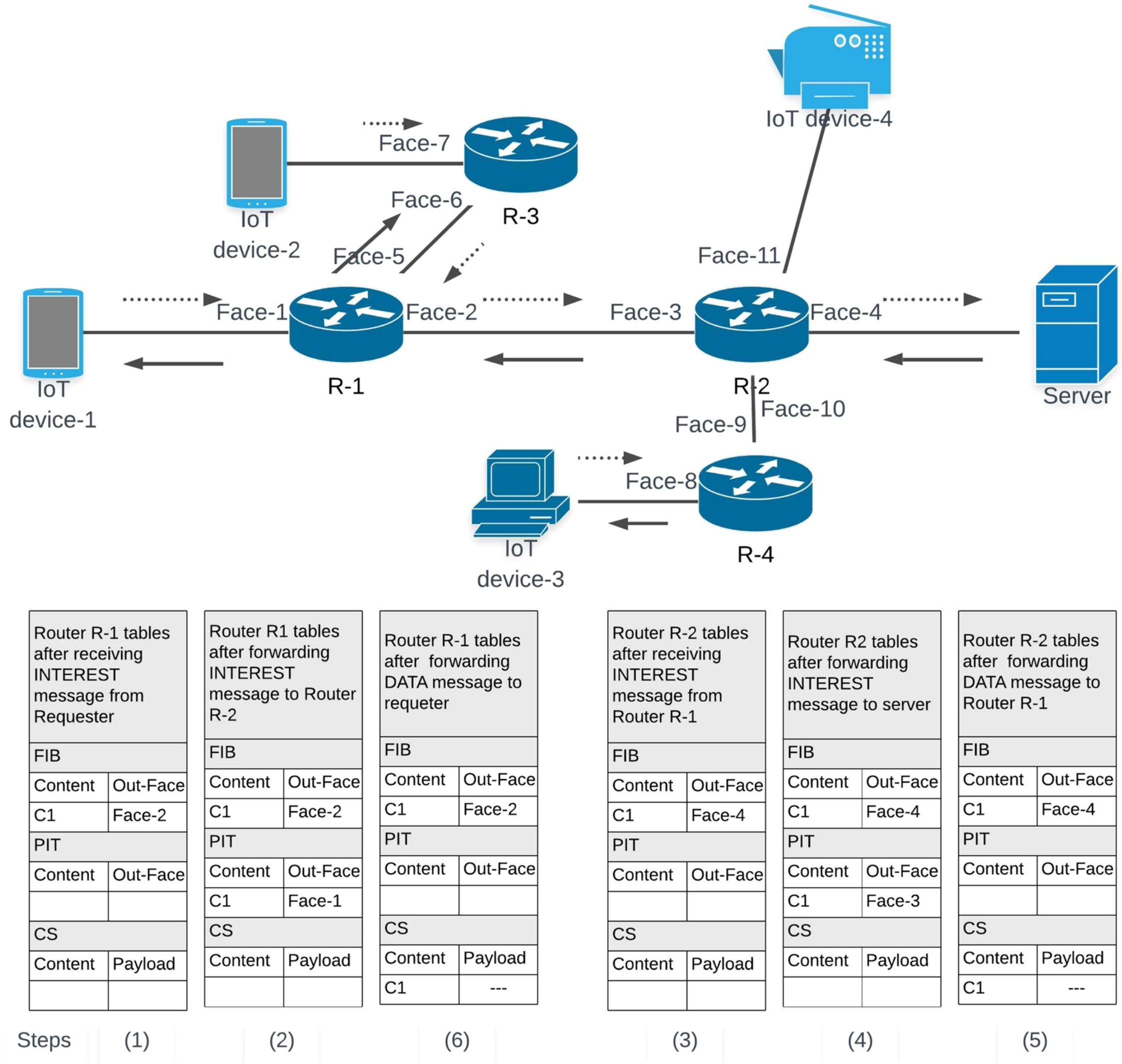
2 Overview of CCN

The CCN is developed as a prospective Internet architecture that alters the host-based Internet to named-content-based Internet architecture. For content name-based information transmission, the CCN provides two types of messages named, Interest messages and Content messages (Wang et al. 2012). The Interest message is used to send the requested content name towards the content provider and the Content message comprises the requested payload. During content delivery, the Content message follows the symmetric path in the backward direction through which the Interest message has arrived.

For content caching and routing operations, each CCN router maintains three data structures named, Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB) as stated in Jacobson et al. (2009). The CS is used to cache the incoming content based on the caching scheme. The PIT stores information of those Interest messages that are forwarded in the network for content retrieval and the content is not received so far. The information of upstream routers is stored in the FIB for Interest message forwarding operations.

Initially, when an Interest message reaches to a router, the router searches its Content Store for the required content. If the required content is present in the CS, then the router/server creates a corresponding Content message and forwards it toward the requester (IoT device). Otherwise, if the content is not present in the CS, then the router explores its PIT for the Interest message. If Interest information is found in the PIT, it indicates that the identical Interest message is already forwarded in the network earlier and its response has not been received. Therefore, the router stores the Interest message along with the existing PIT record and removes the Interest message from the network. If the Interest message information is not present in the PIT, then the router stores Interest message information in the PIT and forward it towards the suitable upstream router according to FIB. If the router is unable to determine the upstream router based on FIB search, then the Interest message is discarded without further dissemination in the network.

An example has been shown in Fig. 1. to demonstrate the states of PIT, FIB, and the CS, of routers during content retrieval in CCN-based IoT systems. Here, the in-network



Router R-1 tables after receiving INTEREST message from Requester	
FIB	
Content	Out-Face
C1	Face-2
PIT	
Content	Out-Face
CS	
Content	Payload

Router R1 tables after forwarding INTEREST message to Router R-2	
FIB	
Content	Out-Face
C1	Face-2
PIT	
Content	Out-Face
C1	Face-1
CS	
Content	Payload

Router R-1 tables after forwarding DATA message to requeter	
FIB	
Content	Out-Face
C1	Face-2
PIT	
Content	Out-Face
CS	
Content	Payload
C1	---

Router R-2 tables after receiving INTEREST message from Router R-1	
FIB	
Content	Out-Face
C1	Face-4
PIT	
Content	Out-Face
CS	
Content	Payload

Router R2 tables after forwarding INTEREST message to server	
FIB	
Content	Out-Face
C1	Face-4
PIT	
Content	Out-Face
C1	Face-3
CS	
Content	Payload

Router R-2 tables after forwarding DATA message to Router R-1	
FIB	
Content	Out-Face
C1	Face-4
PIT	
Content	Out-Face
CS	
Content	Payload
C1	---

Steps	(1)	(2)	(6)	(3)	(4)	(5)

Fig. 1 Example of content retrieval mechanism in content-centric networks

routers (R_1, R_2, R_3, *and* R_4) have the caching capability. The router R_1 communicates with "*IoT device* − 1", routers R_2 and R_3 using interface "*Face* − 1", "*Face* − 2" and "*Face* − 5" respectively. The router R_2 communicates with R_1 and the server using interface "*Face* − 3" and "*Face* − 4" respectively. Initially, the "*IoT device* − 1" generates an Interest message with content (data) C_1 and send it to router R_1. As mentioned in Fig. 1, Router R_1 checks its CS for the requested content in *step* − 1. As the content is not available in the CS of R_1, therefore it searches its PIT for the requested content record. The record does not exist for content C_1 in the PIT because the Interest message has not been forwarded by R_1 earlier. Finally, R_1 lookup its FIB to get information on the next interface (*Face* − 2) to send the Interest message. The R_1 creates a record in its PIT for C_1 and sends the Interest message to R_2 as shown in *step* − 2. Router R_2 receives the Interest using interface "*Face* − 3" and searches its CS and then PIT for the relevant content name in *step* − 3. As the CS and PIT tables are empty for R_2, it creates a record in its PIT for the Interest message along with the content name and the interface information to forward the content after its arrival. Then, it forwards Interest to the server on interface "*Face* − 4" as depicted in *step* − 4. The server prepares the Content message and sends it in the reverse path to router R_2. As shown in *step* − 5, R_2 searches and remove its PIT entry for suitable content and forward the content on "*Face* − 3" towards router R_1. Moreover, R_2 places the content in its cache to satisfy future Interest arrivals (traditional CCN caching strategy). R_1 also performs similar steps as R_2 and forwards the content towards the requester as defined in *step* − 6.

3 Related work

The IoT systems involve a huge diversity of smart devices that are connected with the Internet to access different contents for users (Naeem et al. 2018). In contrast to the conventional devices connected with the IP-based Internet architecture, the IoT devices differ in terms of their communication patterns and infrastructure requirements. The IoT devices require a well-connected network, where devices can distribute the contents with each other in an efficient manner (Adhatarao et al. 2017). Therefore, CCN becomes the most promising Internet architecture for IoT devices that provide name-based access of content copy from the nearest router.

The in-network routers take content placement decisions based on the content caching strategy that reduces content access delay and network traffic significantly. These content caching strategies are generally categorized into two types: the on-path and off-path caching schemes (Khandaker et al. 2019). An on-path caching strategy places the content in the intermediate routers during content delivery while the off-path caching strategy can place the content in any router in the network based on inherent caching criteria (Kumar et al. 2020). However, due to high communication and computational overheads involved in off-path caching strategies (Wang et al. 2015), the on-path caching schemes are extensively used in the CCN-based IoT networks. In recent, various content caching schemes have been proposed to improve network performance for IoT systems.

Jacobson et al. (2009) discuss a content caching scheme named, Leave Copy Everywhere (LCE) that places the incoming content on each on-path router during content delivery. However, due to the higher frequency of caching operations, the scheme suffers from high content replacement operations and computational overhead. The scheme proposed in Laoutaris et al. (2006) suggests that caching the content in selected routers can also improve network performance with reduced caching cost and proposed Leave-Copy Down (LCD) and Move-Copy Down (MCD) caching strategies. In the LCD caching strategy, the content is cached in the immediate downstream router from the router where cache hit occurs, and the remaining intermediate routers forward the content towards the requesters without caching operations. In a variation, the MCD caching scheme works similarly to LCD but the content provider removes the content from its cache after forwarding it to the downstream router. In both, LCD and MCD caching schemes, the contents are placed gradually towards network edges with an increase in content popularity. A random probability-based caching scheme is suggested in Arianfar et al. (2010) that takes random content placement decisions on the intermediate routers and provides rapid caching decision support. The scheme mentioned in Psaras et al. (2012) emphasizes on the caching capacity of the routers and their distance from the content provider for content caching decisions. The scheme heuristically increases the caching probability in those routers that are far from the content provider and are near to the edges of the network. The strategy demonstrated improved network performance as compared to the LCE (Jacobson et al. 2009) and random-probabilistic caching scheme (Arianfar et al. 2010). A probabilistic caching scheme named CPC (Caching Probability Conversion) is proposed in Zhang and Liu (2020) for heterogeneous caching capacities and different content sizes in the IoT networks. The strategy analyzes the cache hit probability and solves the optimization problem using Lagrange multiplier methods.

Several caching schemes also incorporated graph (network) characteristics to improve QoS parameters (Kumar and Tiwari 2020a). The caching strategy discussed in Sourlas et al. (2009) proposed a publisher-subscriber based-approach, in which the content publishers proactively publish the content before its Interest arrival and the caching operations have been performed in the leaf routers to place the contents near to content requesters. Various router centrality-based caching schemes are proposed in Rossi and Rossini (2012) that analyze the caching performance using centrality metrics such as node degree, betweeness, stress, and Eccentricity centrality. The authors have considered the heterogeneous caching capacity of different routers based on the router's importance in the network. The paper concluded that the Degree Centrality (DC) based caching scheme is the simplest and most effective metric for optimal caching operations where the routers with higher degree centrality have more probability of caching the incoming content. The scheme also suggests that the heterogeneous cache size has shown a negligible effect on network performance as compare to homogeneous caching resources. In this direction, our prior work, called CPNDD (Kumar and Tiwari 2020a) jointly considers the node degree centrality and the hop count parameters for the content placement decisions. The scheme suggests increasing the caching probability for higher centrality routers with an increase in the number of hops traversed by the content. Although the schemes demonstrated improved network performance from the LCE and peer caching strategies, the autonomous caching decisions increase content redundancy and lead to poor utilization of network resources. The CSDD (collaborative Caching Strategy Distance and node Degree) scheme (Jaber and Kacimi 2020) analyses the collaborative effect of distance and node degree parameters on caching performance. The scheme suggests to always cache the content in the edge routers if it has not been cached in the intermediate routers. In the scheme, caching probability increases with an increase in node degree and distance parameters. Thus, based on the existing research (Psaras et al. 2012; Kumar and Tiwari 2020a; Jaber and Kacimi 2020), it has been observed that the distance between the content provider and the requester plays a crucial role for efficient cache space utilization and reduces content retrieval delay in the network as content is cached near the requesters with higher caching probability.

A popularity-based caching scheme called MPC (Most Popular Content Caching) has been suggested in Bernardini et al. (2013) that takes caching decisions based on the content popularity. In MPC, the content popularity has been determined based on the Interest count for the cached contents. When a content becomes popular, then it is suggested to the adjacent routers for caching that take autonomous content placement decisions based on their caching strategy. The MPC scheme has been further improved in Ong et al. (2014) to solve the issues related to slow convergence rate and popularity threshold determination. The MAGIC (MAx-Gain In-network Caching) (Ren et al. 2014) caching strategy increases the caching probability for those contents that are frequently accessed by the requesters and are far from the content provider.

The router location and content popularity-based caching scheme is proposed in Wu et al. (2019) that determines the content popularity after analysing the entire network. However, determination of content popularity may not be practicable in large networks. To improve utilization of available caching capacity, previously a scheme called DPWCS (Kumar and Tiwari 2020b) has been proposed by us that effectively places the contents in the network routers. The scheme considers a popularity window to determine popular content during caching decisions. In our prior work by co-author Tiwari et al. Kumar and Tiwari (2021), we have proposed a caching scheme called PDC (Popularity-window and Distance-based efficient Caching) that jointly considers the content popularity and the location of intermediate routers for the caching decisions. However, due to the autonomous caching decisions based on the mentioned parameters, the scheme suffers from a high content replacement rate for the popular contents in the network that increases content retrieval delay. The mutual issues in the mentioned popularity-based caching schemes are the computational complexity and cache space overheads involved during content popularity determination. Due to these concerns, the content popularity-based parameters are not considered during the modelling of the proposed caching scheme in order to reduce computational requirements for the network routers.

Moreover, autonomous caching operations are performed by network routers in the majority of the above-mentioned caching schemes (Jacobson et al. 2009; Arianfar et al. 2010; Rossi and Rossini 2012; Ong et al. 2014; Kumar and Tiwari 2020a, 2021) without cooperation in the network. In collaboration-based strategies, the routers take content caching decisions in collaboration with each other to optimize QoS in the entire network. Therefore, the autonomous content caching schemes limit the network performance to a certain level, and the collaboration-based caching schemes can comprehensively improve QoS in terms of content redundancy, delay, and network utilization (Wang et al. 2015, 2013). A collaborative caching scheme named, greedy caching is proposed in Banerjee et al. (2018), which caches the content using its relative popularity based on content miss rate in the downstream routers. The issue with collaboration-based caching schemes is the additional communication overhead due to cooperation-related messages and scalability concerns (Dai et al. 2012).

To reduce content redundancy and network traffic, several network partitioning-based collaborative caching schemes are proposed in various research papers such as Yan et al. (2017) and Detti et al. (2018). A two-level network planning (Ma et al. 2014) has been recommended in Ma et al. (2017) for load balancing. In Ma et al. (2017), it has been suggested to implement cost-efficient planning in the upper layer and the QoS objectives in the bottom layer of the network. Yan et al. (2017) proposed a two-level hierarchical network partitioning and the content caching decisions take place based on the correlation between the content popularity and router's centrality. However, only a subset of routers perform content caching operations in the network and the remaining routers are used to just forward the incoming contents. This leads to suboptimal network performance due to unused cache memory.

The partitioning scheme discussed in Detti et al. (2018), provides a load balancing mechanism within the partitions while the partitions are visible as a single ICN router to external partitions. In Hasan and Jeong (2018), the authors suggest performing network partitioning based on the number of hops among routers where each partition contains those devices that are one-hop away from each other. However, due to formations of such small partitions, the network may suffer from a large number of network partitions. The network partitioning scheme mentioned in Hasan and Jeong (2019), suggests creating a fixed number of partitions in the network and take content placement decisions based on the popularity of the contents. The summary of the existing content caching strategies has been illustrated in Table 1.

To summarize, many existing research works have explored different parameters and network partitioning models for content caching decisions in Content-Centric Networks. However, to the best of our understanding, no prior work explored the network characteristics-based partitioning along with content provider distance for the content caching decisions. Motivated by the above-mentioned concerns, this paper proposes a novel collaboration-based content caching scheme for optimal performance in IoT networks.

4 System model and assumptions

Let $G(V, E)$ be a network configuration, where V defines various network nodes comprised of IoT devices, routers, and the content provider. The IoT devices (end-users) are represented as $U = \{U_1, U_2, \dots, U_{|U|}\}$, where, U_i denote ith IoT device and $|U|$ defines the total number of IoT devices in the network. The set of in-network routers is represented

Table 1 Comparison of on-path content caching schemes

Author and year	Content placement criteria	In-network hit ratio	Average hop-count	Average network delay	Average network traffic	Network partitioning	Content redundancy
Jacobson et al. (2009)	Cache in each router	Low	High	High	High	No	High
Arianfar et al. (2010)	Random	Low	High	High	High	No	High
Psaras et al. (2012)	Distance, cache size	Moderate	Moderate	Moderate	Moderate	No	High
Rossi and Rossini (2012)	Router's centrality	Moderate	High	High	High	No	High
Bernardini et al. (2013)	Content popularity with static threshold	Moderate	Moderate	Moderate	Moderate	No	High
Ong et al. (2014)	Content popularity with static/dynamic threshold	Low	High	High	High	No	High
Ren et al. (2014)	Content popularity and distance	High	Moderate	Moderate	Moderate	No	High
Kumar and Tiwari (2020a)	Node degree and distance	Moderate	High	Moderate	Moderate	No	High
Jaber and Kacimi (2020)	Centrality and hop-count	High	Moderate	Moderate	High	No	Moderate
Kumar and Tiwari (2021)	Content popularity and distance	High	Moderate	Moderate	High	No	High
Yan et al. (2017)	Content popularity and router centrality	Moderate	Moderate	Moderate	Moderate	Yes	Moderate
Detti et al. (2018)	Hash-based	Moderate	Moderate	Moderate	Moderate	Yes	Low
Hasan and Jeong (2019)	Content popularity	High	Moderate	Moderate	Moderate	Yes	Moderate
Proposed caching scheme	Network partitioning and distance	High	Low	Low	Low	Yes	Low

as $R = \{R_1, R_2, \dots, R_{|R|}\}$, where R_i symbolizes ith router. The content server is represented as $serv$ and therefore, $V = \{U, R, serv\}$. The server contains all distinct contents (content catalogue), which can be requested in the network and work as a sink for all Interest messages. The symbol E defines the set of network links for connection among network devices (V). The network offers multi-hop communication characteristics, in which the Content messages traverse multiple intermediate routers between the content providers and the requesters during content delivery. The content catalogue is represented as $D = \{D_1, D_2, \dots, D_{|D|}\}$, where each distinct content has been defined as D_i and $|D|$ denotes the content catalogue size. The network routers (R) support caching capabilities and can cache the incoming Content messages based on the proposed caching mechanism. For convenience, Table 2 specifies the variable definitions used in the rest of the paper.

To model Interest messages arrival pattern in the network, the proposed scheme considers Zipf distribution for the contents. The Zipf distribution is widely used to simulate content access frequencies for the current Internet applications (Rossi and Rossini 2012; Ren et al. 2014; Shan et al. 2019; Jaber and Kacimi 2020). In Zipf distribution, the popularity of the contents is determined based on their rank and the value of popularity skewness parameter (α). The popularity skewness parameter shapes the content access pattern and models the number of requests to a subset of the content catalogue. The higher value of α narrows the majority of Interest messages to a smaller subset of contents as compared to the scenario where α is too small. Therefore, a larger caching capacity is required to satisfy the Interest messages when the value of α is smaller.

5 Hierarchical partitioning scheme

The proposed caching scheme partitions the IoT network based on hierarchical partitioning, similar to the clustering mechanism discussed in Yan et al. (2017). In the proposed partitioning scheme, the IoT system has been divided into three layers, named backbone-layer, intermediate-layer, and IoT requester-layer. The backbone-layer and the intermediate-layer contain network routers with content caching capability. The IoT requester-layer is comprised of IoT devices that generate content requests and do not have caching resources.

The proposed network-partitioning scheme partitioned the entire network based on the connections among backbone-layer and intermediate-layer routers as shown in Fig. 2. Here, the intermediate-layer routers that are adjacent to a backbone-layer router, are placed within a partition. The backbone-layer router is designated as the partition head and works as a gateway to connect its intra-partition routers to IoT devices, other routers, and the content server. When an intermediate-layer router is adjacent (directly connected) to more than one backbone-layer routers then it would analyse the distance of backbone-layer routers from the server and the partition size for merging decisions. In this scenario, the intermediate-layer router is merged with that backbone-layer router which is closer to the server to reduce content retrieval delay for IoT requester layer. If more than one

Table 2 Variables definition

Variable	Definition
$\lvert D\rvert$	Size of content catalogue
$C(R_i)$	Cache size of Router R_i
$CS(R_i)$	Content Store of router R_i
$C\_name(I_m)$	Requested content name in the Interest message I_m
$C\_name(D_m)$	Content name in the Content message D_m corresponding to the Interest message I_m
λ	Request rate
$\phi_{R_i}^{I_m}$	Boolean variable to determine if content exists in the $CS(R_i)$ for Interest message I_m
T_{R_i}	The threshold value for content caching in R_i
α	Zipf popularity skewness parameter
$PH(R_i)$	Partition head name of router R_i
$S.PH(R_j)$	Partition head name of the content provider (R_j)
$O.PH(R_i)$	Partition head name of the on-path router (R_i) traversed by the Content message
$Dist_{R_j}^{U_k}(I_m)$	Number of hops traversed by Interest message I_m from IoT device (U_k) to router R_j
$Dist_{R_i}^{R_j}(D_m)$	Number of hops traverse by Content message D_m from the content provider (R_j)/ server to intermediate router R_i
$Norm\_dist(R_i,\ R_j,\ U_k)$	Normalized number of hops traversed by the Content message from content provider (R_j) to on-path router (R_i)

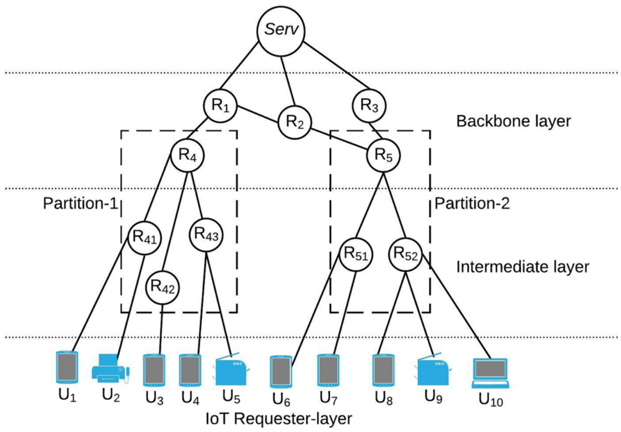


Fig. 2 System design and hierarchical partitioning-based IoT network

backbone-layer router has an identical distance from the server, then adjacent intermediate-layer router is merged in the partition that has lesser number of intra-partition routers to improve load balancing in the network. If a backbone router is not adjacent to any intermediate-layer router, then the backbone router would designate itself as an autonomous partition and the partition head. These backbone routers autonomously cache the incoming contents based on the caching strategy. The partition heads ensure that at most one copy of the incoming Content message can be cached within a partition during inter-partition communication and no content caching operation is performed when the content is requested within the partition by the IoT devices. Therefore, the content redundancy and cache replacement operations would be reduced significantly within partitions and the available caching resources would be utilized compressively.

6 Hierarchical partitioning-based proposed caching scheme

The content provider distance-based metric along with network partitioning has been proposed for content caching decisions in order to improve the QoS for IoT devices. The normalized distance ($Norm\_dist(R_i,\ R_j,\ U_k)$) between the content provider ($R_j/serv$) and the intermediate router (R_i) is determined as the ratio of the hop-count traversed by Content message (D_m) from R_j to R_i and hop-count traversed by Interest message (I_m) from requester IoT device (U_k) to the content provider (R_j). In other words,

$$Norm\_dist(R_i,\ R_j,\ U_k) = \frac{Dist_{R_i}^{R_j}(D_m)}{Dist_{R_j}^{U_k}(I_m)} \tag{1}$$

In the proposed scheme, the content caching decisions take place using the network partitions information and normalized distance metric. Hence, the Interest and Content messages structures have been modified to capture this information as discussed below.

6.1 Modified structure of Interest message

The below-mentioned novel field is appended with the Interest message, which is used for content placement decisions in the proposed caching scheme. The $C\_name(I_m)$ field defines the requested content name in the Interest message. The $Dist_{R_j}^{U_k}(I_m)$ field stores the number of hops traversed by I_m from requester U_k to router R_j.

$C\_name(I_m)$	$Dist_{R_j}^{U_k}(I_m)$	...

6.2 Modified structure of Content message

For effective content placement decisions, the Content message structure has been updated with the below-mentioned fields. Here, $C\_name(D_m)$ defines the content name corresponding to the Interest message I_m. The $Dist_{R_j}^{U_k}(I_m)$ and $Dist_{R_i}^{R_j}(D_m)$ fields contain the hop-count traversed by the Interest message to arrive at the content provider (R_j) and the number of hops traversed by the Content message to reach to the intermediate router R_i from content provider (R_j) respectively. The $S.PH(R_j)$ and $O.PH(R_i)$ represents the names of partition heads of the content provider (R_j) and on-path router (R_i) in the Content message respectively. The *Ctrl* field contains a boolean value which is used during caching decisions. The *Payload* field contains the requested data and other fields of the Content and Interest messages remain unchanged.

$C\_name(D_m)$	$Dist_{R_j}^{U_k}(I_m)$	$Dist_{R_i}^{R_j}(D_m)$	$S.PH(R_j)$
$O.PH(R_i)$	*Ctrl*	*Payload*	...

6.3 Interest message processing mechanism

When an IoT device (U_k) needs to access a content (D_m), it generates the corresponding Interest message (I_m) with $C\_name(I_m)$. The U_k sets the $Dist_{R_i}^{U_k}(I_m)$ field to 0 in the I_m and transmit it towards the content server (Serv). On receiving this Interest message, each on-path router (R_i) performs the steps mentioned in Algorithm 1 (Interest message processing). As stated in *step-1* of the algorithm, after receiving the Interest message, each on-path router (R_i) increases the hop-count traversed by the I_m by 1 in the $Dist_{R_i}^{U_k}(I_m)$ field.

Then, the router searches its CS for the content and determine the value of $\phi_{R_i}^{I_m}$ as follows:

$$\phi_{R_i}^{I_m} = \begin{cases} \mathit{True}, & \mathit{If}\ D_m\ \mathit{exists\ in\ the}\ CS(R_i) \\ \mathit{False}, & \mathit{Otherwie} \end{cases} \tag{2}$$

If the value of $\phi_{R_i}^{I_m}$ is *TRUE* in *step-2* of the algorithm, it implies that the requested content exists in the $CS(R_i)$ and the router then follows Algorithm 2 (Content message processing) to forward the corresponding Content message (D_m) towards the requester. Otherwise, the Interest message is processed using the traditional CCN Interest processing mechanism (Jacobson et al. 2009). As mentioned in *step-3*, the intermediate router R_i checks its PIT for the requested content name and if the record exists then the incoming Interest message (I_m) is aggregated with the existing record in the PIT and further forwarding of the Interest message is stopped. If no record exists in the PIT, then the router forwards the Interest message to a suitable upstream router/server based on the FIB records as defined in *step-4* and creates an entry in the PIT after Interest message forwarding. If no suitable router information exists in the FIB, then R_i removes the Interest message from the network (*step-5*) without further forwarding.

```
Algorithm-1: Interest message processing ($I_m, R_i, U_k$)
1. Update $Dist_{R_i}^{U_k}(I_m)$ field of $I_m$ as
    (a) $Dist_{R_i}^{U_k}(I_m)$= $Dist_{R_i}^{U_k}(I_m)$+1
2. If $\phi_{R_i}^{I_m}$=TRUE, then follow algorithm − 2 mentioned in the next
   section (Content message processing mechanism).
3. Else, If PIT of $R_i$ contains $C\_name(I_m)$, then aggregate $I_m$ in PIT
   and discard $I_m$ from the network.
4. Else, if FIB of $R_i$ contains suitable upstream router information for
   processing of $I_m$, then forward $I_m$ towards the mentioned router.
    (a) Make entry of $I_m$ in the PIT.
5. Else, remove $I_m$ from the network.
```

6.4 Content message processing mechanism

If the Interest message (I_m) reaches to the server or $\phi_{R_i}^{I_m}$ is TRUE for any intermediate router R_j (content provider in this case), then the server/ R_j create a Content message (D_m) as defined in the *step-1* of *algorithm* − 2 (*Content message processing*). The hop-count value mentioned in the $Dist_{R_j}^{U_k}(I_m)$ field of the Interest message is replicated to the D_m and remain constant during content forwarding. The content provider then initialize the value of $Dist_{R_j}^{R_j}(D_m)$ to 0 in the Content message and initialize the partition-head name of the content provider ($PH(R_j)$) in the $S.PH(R_j)$ and $O.PH(R_j)$ fields of D_m as discussed in *step-1(d)*. The *Ctrl* field is set to 0, to control caching decisions within the partitions. The content provider/server then transmits the Content message towards the requester using the same face on which the Interest message is arrived.

```
Algorithm-2: Content message processing ($D_m$, $R_i$, $R_j$, $U_k$)
1. If $R_j$ is content provider then
    (a) Prepare Content message ($D_m$) with the requested payload.
    (b) Replicate $Dist_{R_i}^{U_k}(I_m)$ from the Interest message to $D_m$.
    (c) Initialize, $Dist_{R_i}^{R_j}(D_m)$=0.
    (d) Set $S.PH(R_j)$ and $O.PH(R_j)$ fields in the Content message
        ($D_m$) to $PH(R_j)$.
    (e) Reset the Ctrl field to 0.
    (f) Forward $D_m$ towards $U_k$.
2. Else, Set $Dist_{R_i}^{R_j}(D_m)$=$Dist_{R_i}^{R_j}(D_m)$+1.
3. If, on-path router $R_i$ is a partition head then
    (a) Set $O.PH(R_j)$ field to $R_i$.
    (b) If $S.PH(R_j)! = O.PH(R_i)$ then
            Set the Ctrl field to 1.
4. If $Ctrl$=1 & $O.PH(R_j) = PH(R_i)$ & $S.PH(R_j)! = PH(R_i)$ &
   $Norm\_dist(R_i, R_j, U_k) \geq T_{R_i}$, then
    (a) Cache content in the $CS(R_i)$ using the LRU replacement
        mechanism.
    (b) Reset the Ctrl field to 0.
5. Forward content towards IoT device $U_k$.
```

On arrival of the incoming Content message (D_m), the intermediate router (R_i) performs *step-2 to 5* as illustrated in *algorithm-2*. In *step-2*, the R_i first increases the hop-count traversed by the D_m by 1 in the $Dist_{R_i}^{R_j}(D_m)$ field. As mentioned in *step-3*, if the on-path router R_i is a partition-head then it updates the $O.PH(R_j)$ field of the Content message with its name and set the *Ctrl* field to 1 if its name is different than the partition-head name mentioned in $O.PH(R_j)$ of D_m. These operations enable the content caching within the on-path partition.

During content placement decision, the on-path router R_i ensures that the value of *Ctrl* field is 1 in D_m along with $O.PH(R_j) = PH(R_i)$, $S.PH(R_j)! = PH(R_i)$ and $Norm\_dist(R_i, R_j, U_k) \geq T_{R_i}$. Here, the conditions, $O.PH(R_j)) = PH(R_i)$ and $S.PH(R_j)! = PH(R_i)$ ensure that the at most one copy of the incoming content can cache within each partition. The $Norm\_dist(R_i, R_j, U_k)$ value has been determined using equation 1 and compared with the threshold value T_{R_i}. The optimal value of T_{R_i} is determined empirically based on the network topology. If the above-mentioned conditions are satisfied, then R_i cache the incoming Content D_m using the LRU replacement policy (Laoutaris et al. 2006), when no space is available in the $CS(R_i)$. When the content is cached, the *Ctrl* field in the D_m is reset to 0 to ensure that no further caching operation performed for this Content message within the same partition. After the caching decisions, the content is forwarded to the downstream routers towards the requester (U_k) as shown in *step-5*.

7 Performance evaluation

The performance of the proposed caching scheme has been examined on Abilene network topology, which is a standard network setup for learning purposes in the United States of America (USA) (Alderson et al. 2005). During simulations, the Abilene network contains 167 network nodes that comprise one content server, 33 routers, and 133 requesters (IoT devices). For network partitioning, the routers that are at 1-hop distance from the requester IoT devices are designated as Intermediate-layer routers and the remaining routers become backbone-layer routers, where both layers have the caching capability. The intermediate-layer and backbone-layer routers at 1-hop distance from each other collaborate with each other to form the partitions in the network.

Initially, the entire content catalogue is cached in the content server and the caches of network routers are empty. To study realistic simulation results, the ratio of the cache size ($C(R_i)$) and content catalogue size ($|D|$) has been maintained as 1–2% (Yan et al. 2017). Since no consensus is made for the popularity skewness parameter value (α) in the existing literature and it commonly varies between 0.6 and 1.0 in various research papers (Ren et al. 2014; Lim et al. 2014; Nguyen et al. 2019a), therefore α is set to 0.7 during simulations for realistic network performance. The attributes values used during simulation executions are summarized in Table 3.

Table 3 Simulation parameters values

Parameter	Value
Cache size ($C(R_i)$)	50/100 Contents
Content catalogue size ($|D|$)	5000
Request rate (λ)	50/s per IoT device
Link delay	10 ms
Zipf skewness parameter (α)	0.7
Payload size	1 KB
Link bandwidth	10 Mbps
Simulation duration	1050
Cache replacement scheme	Least Recently Used (LRU)

7.1 Determination of optimal threshold value (T_{R_i})

For the comprehensive utilization of the available cache resources in the proposed caching scheme, the threshold value is determined based on the empirical study in the Abilene network topology. Although, the mentioned heuristics are comparatively arbitrary and may vary for other network setups. However, it provides a good starting point for realistic CCN-based IoT networks. Here, the hit-ratio of the proposed content caching scheme has been determined for different values of (T_{R_i}) with simulation parameters mentioned in Table 3, where the caching capacity of each router is 50 contents. The simulation results analyzed for 1050 STU (Simulation Time Unit) and the average hit-ratio is plotted in Fig. 3.

As illustrated in Fig. 3, the optimal value of cache-hit ratio has been achieved by the proposed scheme, when (T_{R_i}) is 0.5. Similarly, when the cache size is increased to 100, the optimal hit-ratio is attained with (T_{R_i}) = 0.5. Thus, the proposed caching scheme uses a threshold value of 0.5 for content placement decisions during the simulation executions.

7.2 Performance evaluation: network hit-ratio

The average hit-ratio is a critical metric to assess the effectiveness of content caching schemes. A cache hit occurs when the requested content is found in the router. Otherwise, when the requested content is not present in the cache then this occurrence is called a cache miss. The hit ratio of a network is determined as the ratio of the sum of occurrences of cache hit and the sum of occurrences of the Interest messages encountered by the routers. The higher cache hit-ratio indicates that the content placement decisions have been taken efficiently and the users are accessing the content copy from relatively closer routers. Equation 3 formulate the cache hit-ratio metric as follows:

$$Avg\_hit\_ratio = \frac{\sum_{i=1}^{|R|} \left(Cache\_Hit_{R_i} \right)}{\sum_{i=1}^{|R|} \left(Cache\_Hit_{R_i} \right) + \sum_{i=1}^{|R|} \left(Cache\_Miss_{R_i} \right)} \tag{3}$$

Here, $|R|$ represents the number of routers in backbone-layer and intermediate-layer. The $\sum_{i=1}^{|R|} \left(Cache\_Hit_{R_i} \right)$ and $\sum_{i=1}^{|R|} \left(Cache\_Miss_{R_i} \right)$ symbolize the cache-hit and cache-miss occurrences in the entire network in per unit time.

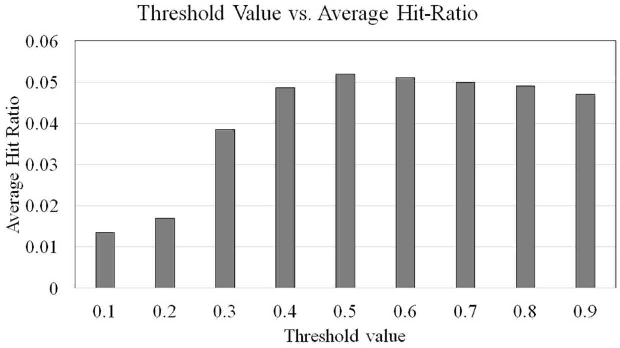


Fig. 3 Optimal threshold value determination (T_{R_i}) for the proposed caching scheme

The performance of the proposed caching scheme has been compared with several standard peer caching schemes with identical network configurations as mentioned in Table 3. When $C(R_i) = 50$, $\lambda = 50/\text{s}$, $|D| = 5000$, and $\alpha = 0.7$, the average hit-ratio observed in the network is plotted in Fig. 4. Initially, all the caching schemes observe relatively low cache hit ratio due to empty in-network caches in the beginning and it increases gradually with time as the contents started being cached in the on-path routers based on the caching strategies. In the simulation results, the proposed scheme outperforms the LCE, LCD, MAGIC, PDC and the CSDD caching schemes by showing 2.9%, 1.9%, 1.2%, 1.1% and 1% gain in hit-ratio respectively.

When caching capacity of in-network routers has been increased to 100 (2% of the content catalogue size) and remaining network configurations remain unchanged as mentioned in the previous simulation scenario, the hit-ratio of the caching schemes increases up to 2.4% as compared to simulation results obtained with $C(R_i) = 50$. Figure 5 illustrates the average network hit-ratio of different caching strategies with cache size=100 and other parameter values as mentioned in Table 3. In this simulation scenario, the proposed scheme demonstrates 1.7% − 3.4% hit-ratio gain from peer caching schemes.

7.3 Performance evaluation: average network hop-count

The average network hop-count metric determines the average hop-count (links) traversed by the Interest messages and the Content messages to deliver the requested contents to the IoT devices in the network. The lower values of this metric specifies that the requested content has been retrieved from a relatively closer router. Therefore, the higher value of average hop-count metric leads to lower QoS for the IoT devices.

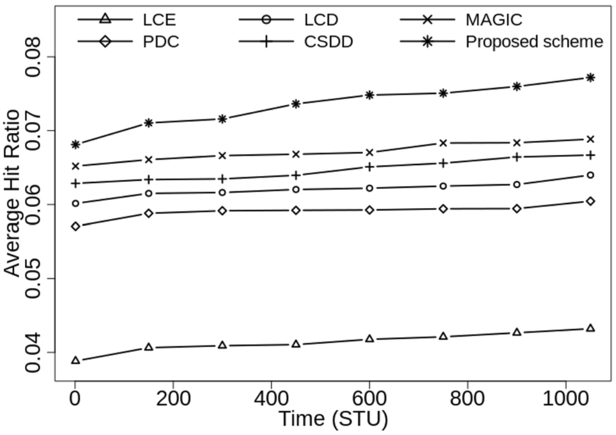


Fig. 5 Average network hit-ratio with $|D| = 5000$, $C(R_i) = 100$, $\alpha = 0.7$ and $\lambda = 50/\text{s}$

Figure 6 demonstrated the average network hop-count observed under different content placement schemes when the router's $C(R_i) = 50$, $\lambda = 50/\text{s}$, $|D| = 5000$, and $\alpha = 0.7$. With these network configurations, the proposed caching scheme reduces the average hop-count by 9.4%, 3.0%, 2.3%, 3.4%, and 2.7% from LCE, LCD, MAGIC, PDC and CSDD caching schemes. The average number of hops traversed to access the contents is higher during the beginning of the simulation and decreases slowly as caching operations take place using content caching schemes.

The average hop-count encountered by Interest/Content messages in all caching schemes decreases significantly, when the size of CS in the routers is doubled to 100 with $\lambda = 50/\text{s}$, $|D| = 5000$, and $\alpha = 0.7$ as shown in Fig. 7. In

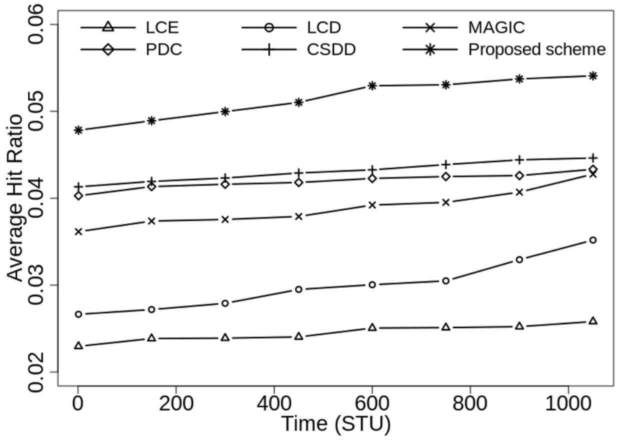


Fig. 4 Average network hit-ratio with $|D| = 5000$, $C(R_i) = 50$, $\alpha = 0.7$ and $\lambda = 50/\text{s}$

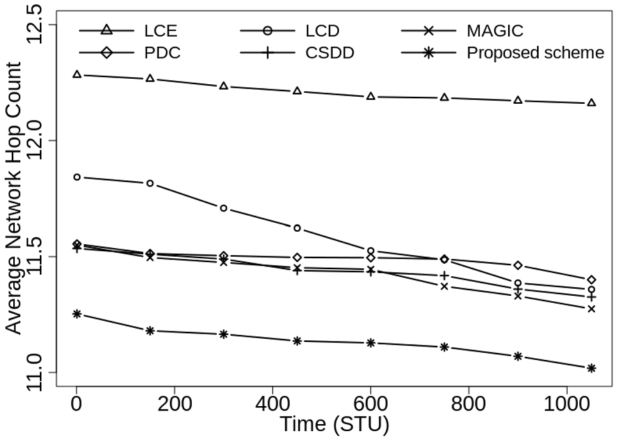


Fig. 6 Average network hop-count with $|D| = 5000$, $C(R_i) = 50$, $\alpha = 0.7$ and $\lambda = 50/\text{s}$

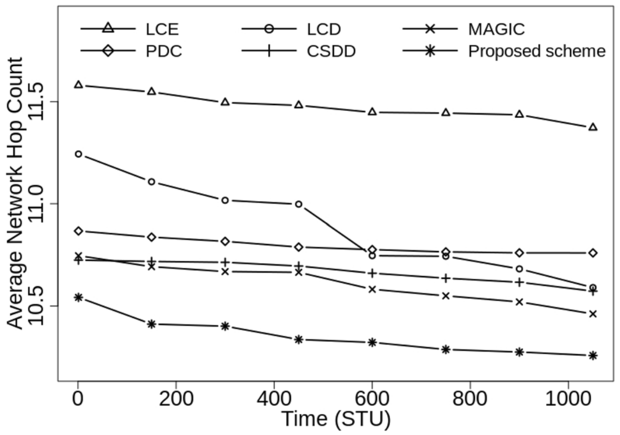


Fig. 7 Average network hop-count with $|D| = 5000$, $C(R_i) = 100$, $\alpha = 0.7$ and $\lambda = 50/s$

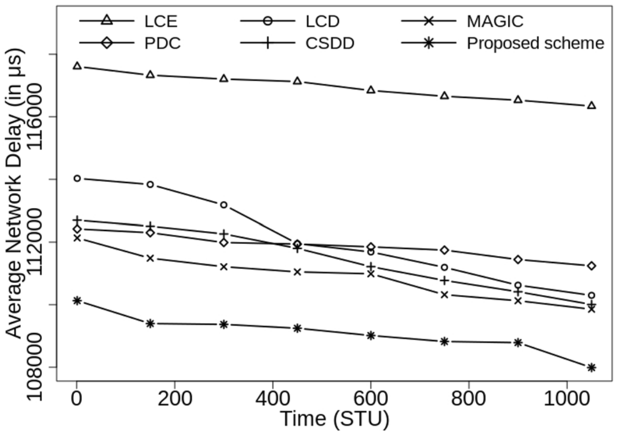


Fig. 8 Average network delay (in µs) with $|D| = 5000$, $C(R_i) = 50$, $\alpha = 0.7$ and $\lambda = 50/s$

this scenario also, the proposed scheme overtakes the existing peer schemes by showing up to 9.8% reduction in average hop-count from peer caching schemes during content delivery. Further, Figs. 6 and 7 demonstrated that caching each content in every on-path router (LCE) does not improve the network performance significantly as compared to heuristic-based caching decisions (LCD, MAGIC, PDC, CSDD, and the proposed caching scheme).

7.4 Performance evaluation: average network delay (in µs)

In this section, the performance of the content caching schemes has been evaluated in terms of average network delay (in µs) encountered by the IoT devices in receiving the requested contents. The network delay is the duration between preparing the Interest message, its first transmission in the network till the delivery of the requested content. The average network delay also incorporates Interest retransmission duration and the computational latency of on-path routers. Hence, this metric provides a comprehensive impression of network performance.

Figure 8 demonstrates the average network delay for various content caching schemes for $C(R_i) = 50$, request rate = 50/s, popularity skewness parameter value=0.7 and content catalogue size = 5000. The simulation results demonstrate the superiority of the proposed caching scheme on existing peer schemes as the proposed scheme reduces average network delay up to 7.2% as compared to existing caching strategies. This indicates that the applied computational heuristics in the proposed caching scheme does not adversely affect the Interest and Content message processing and propagation delay in the network routers as compared to existing schemes.

The average network delay reduces further for all caching schemes when the caching capacity of network routers is increased to 100 with $\lambda = 50/s$, $\alpha = 0.7$, and $|D| = 5000$. As shown in Fig. 9, the average network delay of the proposed caching scheme has been reduced by approximately 750µs as compared to simulation results where the cache size is 50 (shown in Fig. 8). Therefore, it also validates that the performance of the network improves with an increase in the caching capacity of in-network routers. Figure 9 illustrates that the proposed scheme shows approximately 8.6%, 2.5%, 1%, 4.2%, and 2.3% reduction in average network delay as compared to LCE, LCD, MAGIC, PDC, and the CSDD caching schemes for the above-mentioned values of network parameters.

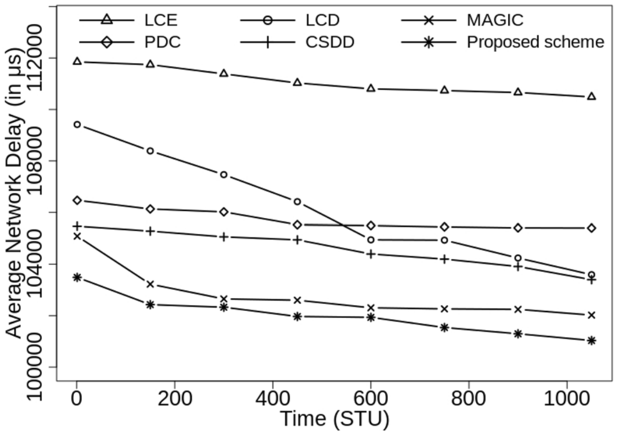


Fig. 9 Average network delay (in µs) with $|D| = 5000$, $C(R_i) = 100$, $\alpha = 0.7$ and $\lambda = 50/s$

7.5 Performance evaluation: average network traffic

The average network traffic metric determines the average traffic load (in Kb/s) on the network links (E) per unit time. The average network traffic indicates the efficiency of the caching scheme in terms of bandwidth consumptions. Therefore, under identical network parameters, the lower network traffic for a caching scheme indicates that the scheme carefully utilizes the available caching resources and reduces the probability of network congestions as compared to other schemes.

The average network traffic observed under different caching schemes has been plotted in Fig. 10 for $C(R_i) = 50$, $\lambda = 50/s$, $\alpha = 0.7$, and $|D| = 5000$. Although the proposed scheme uses a few additional fields in the Interest and Content messages for content placement decisions, the average network traffic has been significantly reduced as compared to LCE, LCD, MAGIC, PDC, and CSDD caching schemes. As illustrated in Fig. 10, the proposed scheme shows up to 5.6% reduction in average network traffic as compared to peer schemes.

Analogous to the previous results, when caching capacity of the network routers increases ($C(R_i) = 100$), the proposed scheme reduces the average network traffic by 7.4%, 3.1%, 1.8%, 3.9% and 2.2% from the LCE, LCD, MAGIC, PDC and CSDD caching schemes respectively. During simulations, the remaining parameters are kept unchanged as shown in Table 3. The results of average network traffic flow for the caching schemes has been depicted in Fig. 11.

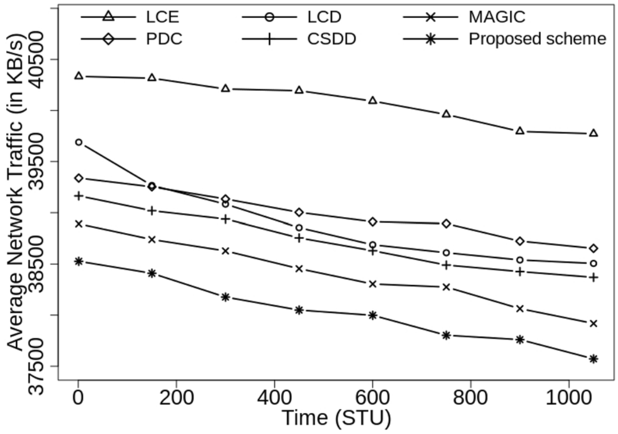


Fig. 10 Average network traffic (in KB/s) with $|D| = 5000$, $C(R_i) = 50$, $\alpha = 0.7$ and $\lambda = 50/s$

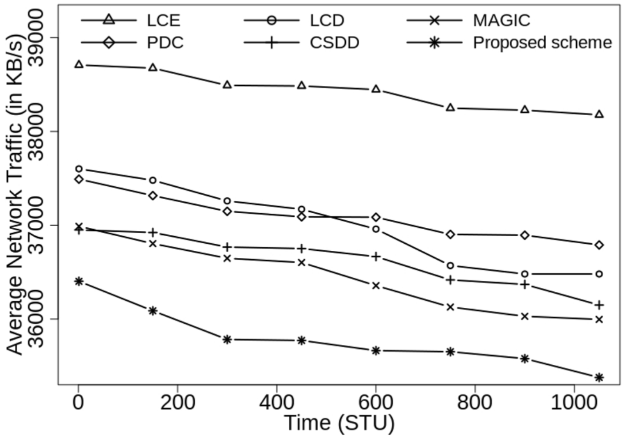


Fig. 11 Average network traffic (in KB/s) with $|D| = 5000$, $C(R_i) = 100$, $\alpha = 0.7$ and $\lambda = 50/s$

7.6 Security aspects of proposed caching scheme

The CCN is vulnerable to several types of attacks that can degrade the effectiveness of in-network caching mechanisms as discussed in Ghali et al. (2014). Some of the common attacks in CCN-enabled IoT networks are related to Interest Flooding Attack (IFA), cache poisoning (storing malicious content in the intermediate routers), and cache pollution (disrupting contents access patterns) (Conti et al. 2013; Guo et al. 2016). The objective of an Interest flooding attack is to overload the PIT by generating poisonous Interest messages for non-existent contents. To mitigate the effect of IFA in the network, the proposed scheme implemented the in-built NACK packet mechanism of the CCN/NDN (Named Data Networking) project (Nguyen et al. 2019b). It is integrated with the reference implementation since NDN version 0.5.1. In order to reduce the network traffic for IFA, the proposed scheme ensure that only one interface would respond to the incoming Interest message instead of forwarding the corresponding content from multiple interfaces. The proposed scheme inherited the "*key locator*" field implemented in the underlying CCN architecture to verify the signature in the Content message. Using this approach, the cache poisoning attack is thwarted by fetching the appropriate key and filtering the poisoned contents.

The cache pollution attacks are generally categorized into two types: false-locality pollution and locality disruption. In false-locality pollution, the attackers continuously generate Interest messages for a few unpopular contents in the network to cache those contents in the network. As the proposed caching scheme considers partitioning and distance-based metrics during content placement decisions instead of the content popularity, thus, the false-locality pollution may not affect the performance of the proposed caching scheme. On the other hand, in locality disruption pollution, the attacker generates

Interest messages for a large number of unpopular contents to increase their caching probability in the network. To identify such malicious users in the network and limiting their traffic, the proposed scheme can be integrated with existing security mechanisms mentioned in Guo et al. (2016), Li et al. (2014) and AbdAllah et al. (2015).

8 Conclusion

In this paper, the potential of the CCN has been elaborated for IoT-based systems. Then, various in-network content caching schemes are analyzed for IoT networks that show improvement in the network performance up to a certain level and raise several limitations on QoS metrics observed by the IoT devices. For effective utilization of the available caching resources, a novel content caching scheme is proposed for IoT systems based on the CCN architecture. The proposed scheme creates hierarchical partitions in the network to reduce content redundancy and takes content placement decisions based on the partition information and content provider distance-based metrics for optimal network performance. Extensive simulations have been performed on the realistic network topology with different network configurations that demonstrated the dominance of the proposed caching strategy on state-of-the-art existing peer schemes. Simulation results show 1.0–3.4% improvement in average network hit-ratio and 2–9.8%, 1–8.6%, and 1–7.4% reduction in average hop-count, delay, and network traffic as compared to LCE, LCD, MAGIC, PDC and CSDD caching schemes for different caching capabilities of in-network routers. The security aspects of the proposed work has also been analysed, which illustrates the robustness of the scheme against known attacks in the CCN-enabled IoT networks. Therefore, the proposed caching scheme is found suitable for CCN based IoT systems for optimal content placement decisions.

In future, the performance of the proposed caching scheme can be investigated in CCN-based 6G architectures and further metrices like popularity, the bandwidth of channels, and congestions can be examined for efficient content placement decisions. Moreover, machine learning and deep learning approaches can be used to select the suitable node for content placements for faster responses in CCN-based IoT networks.

Data availibility statement The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

AbdAllah EG, Hassanein HS, Zulkernine M (2015) A survey of security attacks in information-centric networking. IEEE Commun Surv Tutor 17(3):1441–1454

Adhatarao SS, Arumaithurai M, Fu X (2017) Fogg: A fog computing based gateway to integrate sensor networks to internet. In: 2017 29th International teletraffic congress (ITC 29), IEEE, vol 2, pp 42–47

Alderson D, Li L, Willinger W, Doyle JC (2005) Understanding internet topology: principles, models, and validation. IEEE ACM Trans Netw 13(6):1205–1218

Arianfar S, Nikander P, Ott J (2010) On content-centric router design and implications. In: Proceedings of the re-architecting the internet workshop, pp 1–6

Arshad S, Azam MA, Rehmani MH, Loo J (2018) Recent advances in information-centric networking-based internet of things (ICN-IoT). IEEE Internet Things J 6(2):2128–2158

Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805

Banerjee B, Kulkarni A, Seetharam A (2018) Greedy caching: an optimized content placement strategy for information-centric networks. Comput Netw 140:78–91

Bernardini C, Silverston T, Festor O (2013) Mpc: popularity-based caching strategy for content centric networks. In: 2013 IEEE international conference on communications (ICC), IEEE, pp 3619–3623

Conti M, Gasti P, Teoli M (2013) A lightweight mechanism for detection of cache pollution attacks in named data networking. Comput Netw 57(16):3178–3191

Dai J, Hu Z, Li B, Liu J, Li B (2012) Collaborative hierarchical caching with dynamic request routing for massive content distribution. In: 2012 Proceedings IEEE INFOCOM, IEEE, pp 2444–2452

Detti A, Bracciale L, Loreti P, Rossi G, Melazzi NB (2018) A cluster-based scalable router for information centric networks. Comput Netw 142:24–32

Ghali C, Tsudik G, Uzun E (2014) Network-layer trust in named-data networking. ACM SIGCOMM Comput Commun Rev 44(5):12–19

Guo H, Wang X, Chang K, Tian Y (2016) Exploiting path diversity for thwarting pollution attacks in named data networking. IEEE Trans Inf Forensics Secur 11(9):2077–2090

Hasan K, Jeong SH (2018) A cluster-based content management framework for information-centric networking. In: 2018 Tenth international conference on ubiquitous and future networks (ICUFN), IEEE, pp 891–893

Hasan K, Jeong SH (2019) Efficient caching for data-driven IoT applications and fast content delivery with low latency in ICN. Appl Sci 9(22):4730

Jaber G, Kacimi R (2020) A collaborative caching strategy for content-centric enabled wireless sensor networks. Comput Commun 159:60–70. https://doi.org/10.1016/j.comcom.2020.05.018

Jacobson V, Mosko M, Smetters D, Garcia-Luna-Aceves J (2007) Content-centric networking, whitepaper describing future assurable global networks. Palo Alto Research Center Inc, Palo Alto, pp 1–9

Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp 1–12

Jindal A, Aujla GS, Kumar N (2019) Survivor: a blockchain based edge-as-a-service framework for secure energy trading in sdn-enabled vehicle-to-grid environment. Comput Netw 153:36–48

Khandaker F, Oteafy S, Hassanein HS, Farahat H (2019) A functional taxonomy of caching schemes: towards guided designs in information-centric networks. Comput Netw 165:106937

Kumar N, Iqbal R, Misra S, Rodrigues JJ (2015) Bayesian coalition game for contention-aware reliable data forwarding in vehicular mobile cloud. Future Gener Comput Syst 48:60–72

Kumar S, Tiwari R (2020a) An efficient content placement scheme based on normalized node degree in content centric networking. Clust Comput 24:1277–1291

Kumar S, Tiwari R (2020b) Optimized content centric networking for future internet: dynamic popularity window based caching scheme. Comput Netw. https://doi.org/10.1016/j.comnet.2020.107434

Kumar S, Tiwari R (2021) Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN. Eng Sci Technol. https://doi.org/10.1016/j.jestch.2020.12.018

Kumar S, Tiwari R, Obaidat MS, Kumar N, Hsiao KF (2020) CPNDD: content placement approach in content centric networking. In: ICC 2020-2020 IEEE international conference on communications (ICC), IEEE, pp 1–6

Laoutaris N, Che H, Stavrakakis I (2006) The lcd interconnection of lru caches and its analysis. Perform Eval 63(7):609–634

Li Q, Zhang X, Zheng Q, Sandhu R, Fu X (2014) Live: Lightweight integrity verification and content access control for named data networking. IEEE Trans Inf Forensics Secur 10(2):308–320

Lim SH, Ko YB, Jung GH, Kim J, Jang MW (2014) Inter-chunk popularity-based edge-first caching in content-centric networking. IEEE Commun Lett 18(8):1331–1334

Ma L, Hu K, Zhu Y, Chen H (2014) Cooperative artificial bee colony algorithm for multi-objective rfid network planning. J Netw Comput Appl 42:143–162

Ma L, Wang X, Huang M, Lin Z, Tian L, Chen H (2017) Two-level master-slave rfid networks planning via hybrid multiobjective artificial bee colony optimizer. IEEE Trans Syst Man Cybern Syst 49(5):861–880

Ma L, Cheng S, Shi Y (2020) Enhancing learning efficiency of brain storm optimization via orthogonal learning design. IEEE Trans Syst Man Cybern Syst. https://doi.org/10.1109/TSMC.2020.2963943

Mars D, Gammar SM, Lahmadi A, Saidane LA (2019) Using information centric networking in internet of things: a survey. Wirel Pers Commun 105(1):87–103

Meddeb M, Dhraief A, Belghith A, Monteil T, Drira K (2017) How to cache in icn-based iot environments? In: 2017 IEEE/ACS 14th international conference on computer systems and applications (AICCSA), IEEE, pp 1117–1124

Mishra D, Khan A, Tiwari R, Upadhay S (2018) Automated irrigation system-iot based approach. In: 2018 3rd International conference on Internet of Things: smart innovation and usages (IoT-SIU), IEEE, pp 1–4

Naeem MA, Ali R, Kim BS, Nor SA, Hassan S (2018) A periodic caching strategy solution for the smart city in information-centric internet of things. Sustainability 10(7):2576

Nguyen QN, Liu J, Pan Z, Benkacem I, Tsuda T, Taleb T, Shimamoto S, Sato T (2019a) Ppcs: a progressive popularity-aware caching scheme for edge-based cache redundancy avoidance in information-centric networks. Sensors 19(3):694

Nguyen T, Mai HL, Cogranne R, Doyen G, Mallouli W, Nguyen L, El Aoun M, De Oca EM, Festor O (2019b) Reliable detection of interest flooding attack in real deployment of named data networking. IEEE Trans Inf Forensics Secur 14(9):2470–2485

Nour B, Sharif K, Li F, Biswas S, Moungla H, Guizani M, Wang Y (2019) A survey of internet of things communication using icn: a use case perspective. Comput Commun 142:95–123

Ong MD, Chen M, Taleb T, Wang X, Leung VC (2014) Fgpc: fine-grained popularity-based caching design for content centric networking. In: Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, pp 295–302

Psaras I, Chai WK, Pavlou G (2012) Probabilistic in-network caching for information-centric networks. In: Proceedings of the second edition of the ICN workshop on information-centric networking, pp 55–60

Ren J, Qi W, Westphal C, Wang J, Lu K, Liu S, Wang S (2014) Magic: a distributed max-gain in-network caching strategy in information-centric networks. In: 2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS), IEEE, pp 470–475

Rossi D, Rossini G (2012) On sizing ccn content stores by exploiting topological information. In: 2012 Proceedings IEEE INFOCOM workshops, IEEE, pp 280–285

Shan S, Feng C, Zhang T, Loo J (2019) Proactive caching placement for arbitrary topology with multi-hop forwarding in icn. IEEE Access 7:149117–149131

Shang W, Yu Y, Droms R, Zhang L (2016) Challenges in IoT networking via TCP/IP architecture. In: Technical report NDN-0038 NDN project

Sourlas V, Paschos GS, Flegkas P, Tassiulas L (2009) Caching in content-based publish/subscribe systems. In: GLOBECOM 2009-2009 IEEE global telecommunications conference, IEEE, pp 1–6

Tiwari R, Kumar N (2015) Cooperative gateway cache invalidation scheme for internet-based vehicular ad hoc networks. Wirel Pers Commun 85(4):1789–1814

Tiwari R, Kumar N (2016) An adaptive cache invalidation technique for wireless environments. Telecommun Syst 62(1):149–165

Vural S, Navaratnam P, Wang N, Wang C, Dong L, Tafazolli R (2014) In-network caching of internet-of-things data. In: 2014 IEEE International conference on communications (ICC), IEEE, pp 3185–3190

Wang L, Afanasyev A, Kuntz R, Vuyyuru R, Wakikawa R, Zhang L (2012) Rapid traffic information dissemination using named data. In: Proceedings of the 1st ACM workshop on emerging name-oriented mobile networking design-architecture, algorithms, and applications, pp 7–12

Wang S, Bi J, Wu J (2013) Collaborative caching based on hash-routing for information-centric networking. ACM SIGCOMM Comput Commun Rev 43(4):535–536

Wang S, Bi J, Wu J, Vasilakos AV (2015) Cphr: in-network caching for information-centric networking with partitioning and hash-routing. IEEE ACM Trans Netw 24(5):2742–2755

Wu HT, Cho HH, Wang SJ, Tseng FH (2019) Intelligent data cache based on content popularity and user location for content centric networks. Hum Centric Comput Inf Sci 9(1):1–16

Xylomenos G, Ververidis CN, Siris VA, Fotiou N, Tsilopoulos C, Vasilakos X, Katsaros KV, Polyzos GC (2013) A survey of information-centric networking research. IEEE Commun Surv Tutor 16(2):1024–1049

Yan H, Gao D, Su W, Foh CH, Zhang H, Vasilakos AV (2017) Caching strategy based on hierarchical cluster for named data networking. IEEE Access 5:8433–8443

Zhang S, Liu J (2020) Optimal probabilistic caching in heterogeneous IoT networks. IEEE Internet Things J 7(4):3404–3414

Zhang Z, Lung CH, St-Hilaire M, Lambadaris I (2020) Smart proactive caching: empower the video delivery for autonomous vehicles in ICN-based networks. IEEE Trans Veh Technol 69(7):7955–7965