



Modified Floyd Warshall Algorithm for Cache Management in Information Centric Network

Boddepalli Kiran Kumar^{1*} **Edara Srinivasan Reddy²**

¹*Acharya Nagarjuna University College of Engineering and Technology, India*

²*Acharya Nagarjuna University, Department of Computer Science & Engineering, Guntur, India*

* Corresponding author's Email: kiranbdpl@yahoo.com

Abstract: Information Centric Network (ICN) is one of the growing network environments that provides the majority of internet activities, which are related to information access and delivery. In ICN, users can obtain information contents of the caches in ICN routers. However, how to use the caches effectively is one of the main challenges faced by the researchers. Thus, several research studies are developed for reducing the cache overlaps among routers, since that limits the interests towards the caches. In order to address this problem, a new shortest path estimation algorithm (Modified Floyd Warshall Algorithm (MFWA)) was developed to identify the shortest path in each subnet for retrieving the cache memory data quickly. In MFWA, a new iterative matrix was used to eliminate the invalid paths that helps to retrieve the cache memory data quickly. In this research work, an ICN network was generated based on the concept of GEANT topology and tested with dissimilar cache management policies such as Hash Routing Symmetric (HR-SYMM), Hash Routing Asymmetric (HR-ASYMM), Hash Routing Hybrid Asymmetric Multicast (HR-HYBRID-AM), Hash Routing Hybrid Symmetric Multicast (HR-HYBRID-SM), and Hash Routing Multicast (HR-MULTICAST). In experimental section, the proposed shortest path estimation method showed better performance in light of latency, link load and cache hit ratio as compared to the existing approaches: Floyd Warshall Algorithm (FWA), Dijkstra's Algorithm (DA), and Enhanced Dijkstra's Algorithm (EDA). The experimental outcome showed that the proposed approach improved cache hit ratio up to 4-30% as related to the existing methods.

Keywords: Cache management policies, GEANT topology, Information centric networking, Modified Floyd Warshall algorithm.

1. Introduction

The ICN is an internet architectural model that replaces the aging host based internet architecture with content centered architecture [1]. Presently, ICN broadly varies in design and implementation, but nearly all networks have a few similarities in common that includes content as first class citizen, mobility, content distribution without dependence upon network overlays, in-network caching, decoupling of sender, receiver, content names and locations [2-4]. In order to achieve these goals, several networks are developed like data-oriented network architecture, network of information, Content-Centric Networking (CCN), and named

data networking [5, 6]. The ICN states a research direction of content-information-data-centric approach to the network architecture. Related to traditional internet architectures, the content-centric is very effective in content access and conveyance, with a great interest in volume of digital and media data (i.e., movies, audio files, short videos, images) [7-9]. ICN is a future internet architecture that is directed to disentangle the above-mentioned issues by worldwide content caching in content routers and location independent content naming. In ICN, content name is decoupled from the host address that effectually release the role of an identifier and locator intake apart deviation to present the internet protocol address, which serves for both the purposes [10, 11].

The enhancement in the liberation of a popular content is achieved by content naming that unswervingly leads to the exploitation of in-network caching. The authentication and identification of each content item is done without being concomitant to a specific host. The bandwidth consumption is lowered by caching the application independent content item that is re-accessed by other users. Hence, ICN is a network layer solution, while all the routers in ICN are responsible to store downloaded content for a low time [12-13]. In centric network paradigm, users request the content by utilizing content name and then the network performs mapping between the request and the content location. The experience of end user depends on how fast the requested content is found and delivered, while it is essential to collect the popular content in cache nodes. Hence, the network caching is influenced by business models and operator related control [14]. In this research study, an ICN network was generated on the basis of GEANT topology. It is a pan European data network, which interconnects education networks across Europe and national research. The GEANT topology enables collaboration on projects that ranges from biological science to earth observation, arts and culture. The GEANT project combines high-capacity and high-bandwidth 50,000 kilometre network with a growing range of services. The generated ICN network was tested with five cache management policies such as, HR-SYMM, HR-ASYMM, HR-HYBRID-AM, HR-HYBRID-SM and HR-MULTICAST. After network generation, MFWA was proposed for identifying the shortest path in every subnet and between subnets. Compared to conventional FWA, a new iterative matrix was included in the MFWA for eliminating the invalid paths that helps to retrieve the cache memory data quickly. In MFWA, a single execution was enough to find the length of the shortest paths between all pairs of vertices and also it was easy to modify the algorithm and use it to reconstruct the paths. At last, the proposed method performance was related to the previous methods by means of link load, latency, and cache hit ratio. From the experimental result, the proposed method improved cache hit ratio up to 4-30% as compared to the existing methods.

This research paper is arranged as follows. Numerous papers on cache management in ICN are reviewed in the section 2. Detailed explanation about the proposed methodology is given in the section 3. Section 4 illustrates about the quantitative and comparative analysis of proposed methodology. The conclusion is done in the section 5.

2. Literature review

Numerous methodologies are developed by the researchers in cache management in ICN. In this sub-section, a brief discussion of some important contributions to the existing literatures is presented.

C. Li, S. Gong, X. Wang, L. Wang, Q. Jiang, and K. Okamura [15] used a new Crowd-sourced Vehicular Content-Centric Networking (CVCCN) approach for secure and efficient content distribution. The developed methodology enabled the nearby vehicles in order to crowd-source the radio links and caching resources for cooperative content distribution. The CVCCN methodology formulated the problem as the maximization of all users' payoffs and also developed an online scheduling approach to address the above-mentioned concerns. Further, the developed methodology adopted identity-based proxy re-encryption for securing the process of content distribution. The experimental outcome showed that the developed approach improves the performance in light of average requester utility related to existing approaches. Here, the incentive procedure was essential to motivate vehicles for participating in crowd-sourced content distribution system to help each other. Still, the developed approach needs to concentrate more on content distribution to solve the incentive problems.

J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal [16] developed a new system for optimum cache management in ICNs, which consider both content routing and caching strategy. In this study, the developed system works based on software-defined networking, while a controller was responsible to determine the optimum content routing and caching utilizing linear network coding. Then, a significant Network Coding based Cache Management (NCCM) algorithm was utilized to obtain routing solution and near-optimal caching for ICNs. The developed system further derived the upper and lower bounds and extensive experiments were performed for comparing the performance of NCCM algorithm. In addition to network bandwidth cost, network security was another key concern in communication networks. In this research, the developed methodology failed to explore the search methods over encrypted data for network security.

S. Paul, B. Banerjee, A. Mukherjee, and M. K. Naskar [17] developed a new approach for addressing the content management concern in a cache with finite storage capability with Q-routing. Additionally, a superior content management policy was used to change a router to a self-sustained cache. Moreover, the developed approach effectively

processes the content packets in a cache and reduced the propagation delay through a cache. The developed method was simulated on real-life network environment by means of throughput, latency, goodput, and link load. The simulation results substantiated the improvement in latency performance for dissimilar network parameters and shown significant improvement over the existing methods. Despite of adopting the Q-routing, the developed approach reduced the redundancy of static routing table that may increases the extra computational burden.

M. Bilal, and S.G. Kang [18] implemented a cache eviction and replication schemes: Conditional Leave a Copy Everywhere (CLCE) and Least Frequent Recently Used (LFRU) in ICN. For content spread, the developed system utilized CLCE replication scheme that ensures the content was replicated in the cache node. The developed methodology also utilized the concept of replication priority, which was defined and managed by the local administrator to reflect the operator related control, business model demands, and application specific. The LFRU was implementable for changing the cache networks. Experimental results proved that the cache network with LFRU was ergodic and convergent. The developed methodology also estimated the probability of a hit for LFRU, and finally compared the performance of LFRU with existing systems. However, in this research study, the developed method considered a cache with a limited size and failed to analysis the infinite cache.

J. Shi, X. Wang, and M. Huang [19] developed a cache-aware routing method in Mobile Social Network (MSN) based on ICN paradigm. In the developed work, interest routing method was devised based on social regularity, which were attained by utilizing the content names. Here, Data routing approach was developed based on the closeness metrics among nodes to transfer the data packets to the interest requesters. Experimental consequences showed that the developed scheme has higher message delivery ratio and lower network overhead than other existing schemes. In cache-aware routing, mobile users with same social features were constructed as a community that was considered as one of the main issues in the developed routing scheme.

I. Psaras, W.K. Chai, and G. Pavlou [20] presented an idea of resource management for in-network caching environments. In this research, an algorithm; ProbCache was used for caching the contents on the basis of multiplexes content flows and path lengths. The ultimate goal of ProbCache

was to use resources efficiently, and for reducing the caching and network traffic redundancy. The developed system exhibits better performance by means of network resource utilization and resource allocation among content flows. In experimental phase, the developed system shows fast convergence to cache the popular content items. In the developed system, reusing and caching the previous fetched resources was difficult.

B. Banerjee, A. Seetharam, A. Mukherjee, and M.K. Naskar [21] developed a new routing approach that leverages the concept of characteristics time for improving the latency. The developed routing method utilizes the characteristics time information for forwarding requests to caches. In this research, the Dijkstra's algorithm works with existing cache replacement policy and management, which were implemented in ICN prototypes with minimal effort. Experimental simulation was carried out on different internet topologies (for instance: GEANT). The developed routing algorithm attained good latency compared to the existing works. A major drawback in Dijkstra's algorithm was that it cannot manage the negative cycles that leads to acyclic graphs.

S. Ahmed, R.F. Ibrahim, and H.A. Hefny, [22] developed an enhanced routing technique: Dijkstra's algorithm and Analytical Hierarchical Processing (AHP) for mobile based route network analysis. Initially, classical Dijkstra's algorithm was applied to the weighted road network to give the best path between any two nodes in the network. In this research, eight impedance factors were considered for mobile based route network analysis such as, road length, travel time, road type, road width, velocity limits, traffic volume, mass density, and junction delay. The developed routing approach was implemented in mobile applications, and the performance was compared with other shortest path algorithms. The major drawback of the developed algorithm that it selects more unnecessary resources.

To overcome the above-mentioned problems, a new shortest path estimation algorithm (MFWA) is developed in order to estimate the shortest path in the generated ICN network. This action helps to retrieve the cache memory data quickly and also to improve the cache hit ratio.

3. Proposed methodology

In this research study, the proposed system contains three major phases like network generation, shortest path estimation and cache management. Fig. 1 represents the working procedure of proposed system. Detailed explanation about the proposed system is described below.

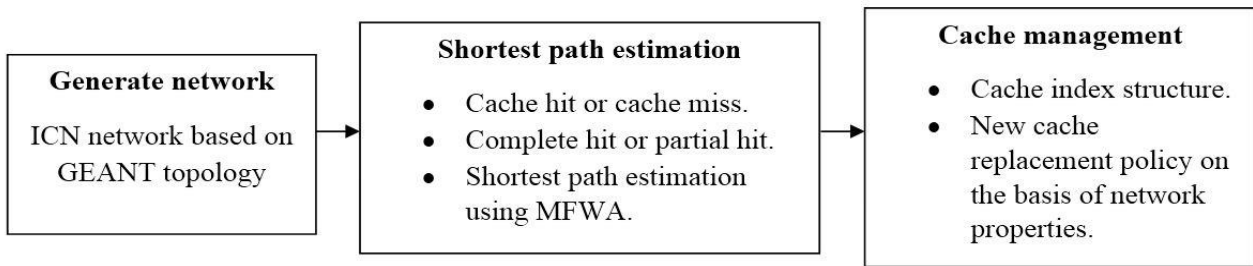


Figure.1 Block diagram of proposed system

3.1 Generating information centric networking based on GEANT topology

Generally, ICN enables new ideas for naming and addressing, privacy, and security. ICN creates current knowledge on the basis of network deployment, operation, management, and borrows elements from previous research studies in distributed systems, application layer caching, transport and network protocol, multicast, and peer-to-peer system. In addition, ICN marks an important shift in communication and networking and also it focuses on finding and transmitting information to end users instead of connecting to end hosts. The key concept of ICN to have a huge influence on the familiar textbook protocol stack. Usually, the network architecture develops new prospects for all stakeholders that comprises of service and content providers, network operators, equipment vendors, and above all end users. In this research, the ICN is generated on the basis of GEANT network, which comprises of 17 nodes and 25 links and the set of data for analysis is collected at the interval of 15 minute per run.

3.2 Shortest path estimation

After generating the network, determine the shortest path for a new query by relating with available queries. Generally, the matched query will be either partial hit or complete hit. In the generated network, partial hit befalls if there exists one queried-path. Successively, the complete hit befalls if there exists one queried-path with a source and destination node pair matching exactly with the input query. In this research, MFWA is utilized for estimating the shortest path in the generated network.

3.2.1. Floyd Warshall algorithm

The Floyd algorithm is utilized for finding the shortest path to retrieve the data from ICN, which is considered as the minimum sum of weighted edges for all paths between the nodes. In this algorithm, distant matrix provides the edge weight for adjacent vertices. If the two vertices are adjacent, then the

corresponding distant matrix entry is infinity. In addition, Warshall algorithm utilizes the adjacency matrix to identify the transitive closure of a directed graph. The combined Floyd and Warshall algorithm is utilized for finding the shortest path in the system either it comprises of negative or positive value [23]. The single cycle of FWA algorithm identifies the shortest path in the vertices.

The FWA is applied to the directed graph that is generated for the data obtain from ICN. At first, generate a network as weighted graph $G = (V, E)$, where V is stated as vertices, which numbered from 1 to N and E is indicated as edge. The edge weight w_{xy} between the vertices x and y in the graph $G = (V, E)$ is evaluated by utilizing the Eq. (1).

$$w_{xy} = \begin{cases} 0 & \text{if } x = y \\ \text{weight of a directed edge}(x, y) & \text{if } x \neq y \text{ and } (x, y) \in E \\ \infty & \text{if } x \neq y \text{ and } (x, y) \notin E \end{cases} \quad (1)$$

The $n \times n$ matrix indicates the edge weights of an n -vertex graph. The shortest path distance between the nodes/vertices x and y with intermediate vertex k in the network $\{1, 2, 3, \dots, k\}$ is signified as $d_{xy}^{(k)}$. Hence, there are two possible ways to find the shortest path distance between the vertices x and y , (i) shortest path length is $d_{xy}^{(k-1)}$, if K is not consider as a vertex, and (ii) shortest path length is $d_{xk}^{(k-1)} + d_{yk}^{(k-1)}$, if K is consider as a vertex. So, the shortest path $d_{xy}^{(k)}$ is evaluated using recursion formula, which is indicated in the Eq. (2).

$$d_{xy}^{(k)} = \begin{cases} w_{xy} & \text{if } k = 0 \\ \min(d_{xy}^{(k-1)}, d_{xk}^{(k-1)} + d_{yk}^{(k-1)}) & \text{if } k \geq 1 \end{cases} \quad (2)$$

Besides, the predecessor of vertex y on a shortest path from vertex x with all intermediate vertices $\{1, 2, 3, \dots, k\}$ is given in the Eqs. (3) and (4).

$$\pi_{xy}^{(0)} = \begin{cases} \text{nil} & \text{if } x = y \text{ or } w_{xy} = \infty \\ x & \text{if } x \neq y \text{ or } w_{xy} < \infty \end{cases} \quad (3)$$

$$\pi_{xy}^{(k)} = \begin{cases} \pi_{xy}^{(k-1)} & \text{if } d_{xy}^{(k-1)} \leq d_{xk}^{(k-1)} + d_{yk}^{(k-1)} \\ \pi_{yk}^{(k-1)} & \text{if } d_{xy}^{(k-1)} > d_{xk}^{(k-1)} + d_{yk}^{(k-1)} \end{cases} \quad (4)$$

At first, FWA applies the shortest path (x, y, k) function to the vertices x to y for the first set $k = 1$, and then evaluated for second set $k = 2$, upto $k = N$.

3.2.2. Modified Floyd Warshall algorithm

The MFWA inspects the all possible paths between the given pair of vertices. At first, MFWA regenerates the paths with minor modification, an iterative matrix is employed in the algorithm in order to eliminate the invalid path. The path between any two vertices (x, y) has another similar path with same weight W in the reverse direction. If there is no reverse path or direct acknowledgement path between the two vertices (x, y) then there will be no direct path exists between the vertices (x, y) . If weight of the path is not equal to reverse path, then the respective path is invalid.

Before estimating the shortest path distance $d_{xy}^{(k)}$ from the vertices (x, y) , consider an ICN as graph $G = (V, E)$, where V is denoted as vertices, and E is represented as edge. Then, validate the path under two conditions, (i) length is equal to the reply path, and (ii) presence of corresponding acknowledgement path. By utilizing this two cases, eliminates the invalid paths between the vertices $d_{xy}^{(l)}$. Then, the shortest path between the vertices x and y is determined for valid paths $d_{xy}^{(V)}$. The mathematical equation for eliminating the invalid paths is denoted in Eq. (5).

$$d_{xy}^{(k)} = \begin{cases} d_{xy}^{(l)} & d_{xy} \neq d_{yx}, Wd_{xy} \neq Wd_{yx} \\ d_{xy}^{(V)} & d_{xy} = d_{yx}, Wd_{xy} = Wd_{yx} \end{cases} \quad (5)$$

After rejecting the invalid paths, shortest path is evaluated among the valid paths $d_{xy}^{(V)}$ using the Eq. (6). Pseudo code of proposed shortest path algorithm (MFWA) is detailed below.

$$d_{xy}^{(k)} = \begin{cases} \min(d_{xy}^{(k-1)}, d_{xk}^{(k-1)} + d_{yk}^{(k-1)}) & \text{if } k \geq 1 \\ w_{xy} & \text{if } k = 0 \end{cases} \quad (6)$$

3.2.2.1. Pseudo code of modified Floyd Warshall algorithm

In this sub-section, pseudo code of MFWA is briefly explained. Related to FWA, the MFWA significantly removes the unwanted or invalid paths. In MFWA, a few changes are carried out in the step 8. If the path is unwanted or invalid, the condition automatically gets break. At last, the loop ends after calculating the shortest path between the vertices using the steps 11 and 12.

1. Let distance be a $|V| \times |V|$ array of minimum distances initialized to ∞ (infinity)
2. For each weighted graph $G = (V, E)$, $E \leftarrow$ edge
3. $dist[i, j] \leftarrow w(i, j)$ // weights of an N-vertex graph
4. For each vertex V
5. $dist[V][V] \leftarrow 0$
6. Fork from 1 to $|V|$
7. For i from 1 to $|V|$
8. If $(dist[i][k] \neq \infty)$
9. For j from 1 to $|V|$
10. If $(dist[k][j] \neq \infty)$
11. If $dist[i][j] > dist[i][k] + dist[k][j]$
12. $dist[i][j] \leftarrow dist[i][k] + dist[k][j]$
13. End
14. End
15. End
16. End
17. End
18. End

3.3 Cache management

After identifying the shortest path in generated network, the unmatched new queries are updated in cache server for future use. Besides, the shortest path is determined for new queries by relating with path patterns. Then, identify the query by using a grid based design, if the new query is unmatched. On the basis of MFWA, determine the shortest path for a new query by relating with available queries. Though, the matched query will be either partial hit or complete hit. Respectively, the unmatched new queries are updated in the cache for future use to attain the path pattern in a timely manner.

4. Experimental result and discussion

In this experimental section, the proposed system was simulated by using python with 3.5 GHz, Intel i5 processor, 4 GB RAM and 1 TB hard disc. The developed GEANT network topology consists

of 17 nodes and 25 links and the set of data for analysis is collected at the interval of 15 minute per run. To evaluate the efficiency of proposed system, the performance of proposed shortest path estimation approach (MFWA) was compared with a few existing shortest path estimation algorithms (FWA, DA [21], and EDA [22]) and also tested with five dissimilar cache management policies like HR-SYMM, HR-ASYMM, HR-MULTICAST, HR-HYBRID-AM, and HR-HYBRID-SM. In this research, the proposed system performance was validated by means of link load, latency, and cache hit ratio.

4.1 Performance measure

Performance metric is defined as the procedure of analysing, reporting, and collecting information about the performance of an individual or group. The detailed explanation about the undertaken performance metric (cache hit ratio, link load, and latency) are given below.

Cache hit ratio: It is the ratio of number of cache hits to the number of lookups, which is exponentially denoted in terms of percentage (%). The formula for calculating a cache hit ratio is mentioned in the Eq. (7).

$$\text{Cache hit ratio} = \frac{\text{Number of cache hits}}{(\text{Number of cache hits} + \text{Number of cache misses})} \times 100 \quad (7)$$

Link load: It is evaluated on the basis of content request rate. The link load is defined as an average of the total number of contents passing through link over per unit run time.

Latency: The latency is defined as the amount of time a message takes to traverse a system. In computer network, latency is an expression of how much time a network takes to transfer a packet from one node to destination node. Usually, latency depends on transmission medium and other devices like routers and modems. A low latency states a high network efficiency.

4.2 Quantitative analysis

In this subsection, the quantitative analysis of existing and proposed work is carried out in terms of latency and link load performance. Here, the efficiency of proposed method is validated with two existing shortest path estimation approaches (FWA, DA [21], and EDA [22]) and five cache management policies (HR-SYMM, HR-ASYMM, HR-MULTICAST, HR-HYBRID-AM, and HR-HYBRID-SM). Also, the proposed and existing shortest path estimation approach is validated with different network caches. Inspecting the Tables 1

Table 1. Quantitative analysis of proposed and existing work in terms of latency

Cache policies	Shortest path estimation algorithm	Latency				
		Network cache				
		0.01	0.05	0.1	0.5	1
HR-SYMM	DA [21]	86.14	78.15	75.20	70.02	69.9
	EDA [22]	79.12	68.84	74.03	54.19	49.62
	FWA	76.24	64.27	57.41	24.85	12.7
	MFWA	91.13	79.19	72.24	39.54	27.5
HR-ASYMM	DA [21]	81.21	79.89	74.57	66.81	63.27
	EDA [22]	80.41	74.90	72.01	62.07	60
	FWA	80.82	75.29	72.40	60.45	51.6
	MFWA	81.54	81.55	81.54	81.55	81.56
HR-MULTICAST	DA [21]	76.24	72.72	57.41	24.85	12.77
	EDA [22]	78.96	74.61	62.92	50.94	44.49
	FWA	81.33	77.02	74.34	62.61	55.54
	MFWA	86	74.90	68.46	37.84	26.7
HR-HYBRID-AM	DA [21]	81.35	76.84	74.16	62.08	54.06
	EDA [22]	81.39	77.91	75	63.03	56.72
	FWA	81.42	78.20	76.18	66.26	58.44
	MFWA	86.04	75.19	69.53	49.14	47.30
HR-HYBRID-SM	DA [21]	81.42	80.57	76.18	66.26	58.44
	EDA [22]	81.20	78.92	75.98	66.40	61.27
	FWA	81.21	76.84	74.57	66.81	63.2
	MFWA	81.09	76.28	73.87	64.1	57.8

Table 2. Quantitative analysis of proposed and existing work in terms of link load

Cache policies	Shortest path estimation algorithm	Link load				
		Network cache				
		0.01	0.05	0.1	0.5	1
HR-SYMM	DA [21]	754.6	669.9	640.60	585.44	584.87
	EDA [22]	754.90	652.09	611.72	361.91	408.26
	FWA	755.34	636.64	567.68	245.6	126.33
	MFWA	755.46	638.26	568.57	244.91	126.95
HR-ASYMM	DA [21]	758.48	711.55	687.14	609.4	574.06
	EDA [22]	756.66	699.71	672.92	549.01	488.74
	FWA	755.01	694.49	665.36	538.8	444.54
	MFWA	760.31	762.41	764.34	761.34	761.47
HR-MULTICAST	DA [21]	759.03	712.92	685.97	563.82	490.27
	EDA [22]	759	712.62	680.2	569.40	487
	FWA	759.03	712.92	685.97	563.82	490.27
	MFWA	754.5	636.50	569.24	244.37	126.60
HR-HYBRID-AM	DA [21]	755.34	636.64	567.68	245.6	126.33
	EDA [22]	754.92	692.86	650.95	412.59	386.41
	FWA	760.63	725.42	703.61	599.84	519.44
	MFWA	755.83	639.26	580.66	362.04	342.28
HR-HYBRID-SM	DA [21]	755.01	694.49	665.36	538.8	444.54
	EDA [22]	756.402	702.82	678.93	587.03	481.19
	FWA	758.48	711.55	687.14	609.46	574.06
	MFWA	756.76	706	681.08	580	513.01

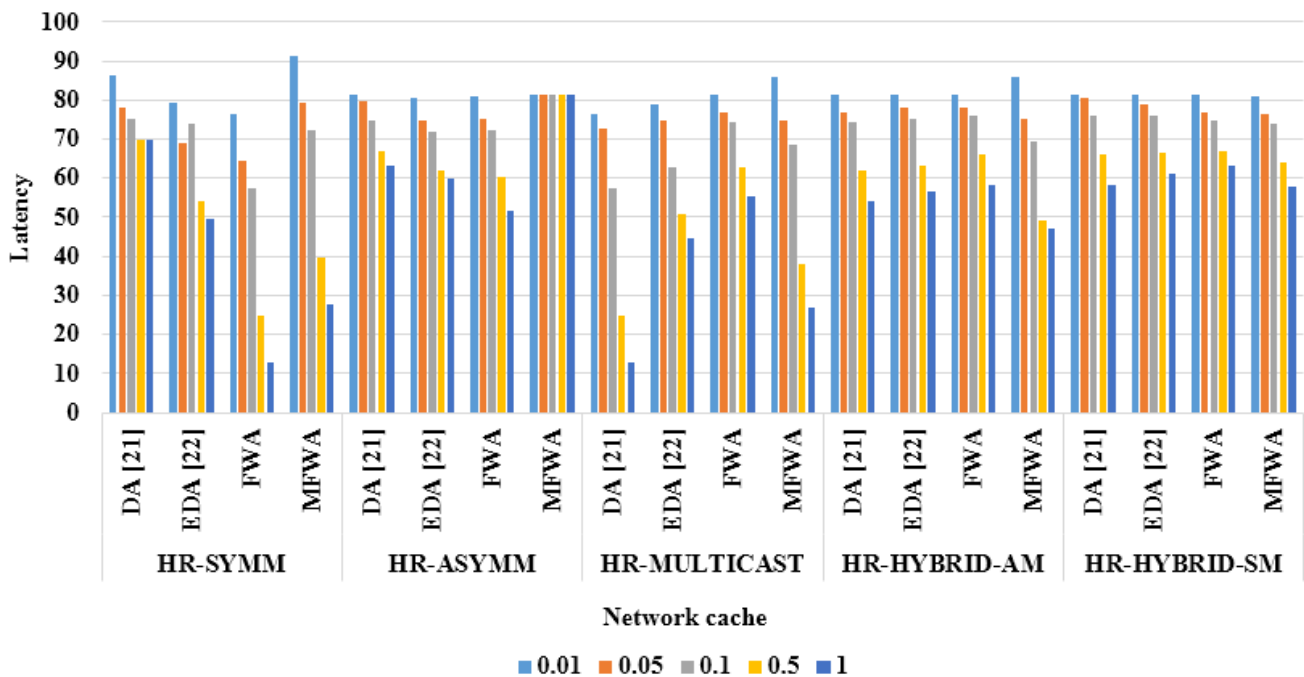


Figure.2 Graphical comparison of proposed and existing methodology in terms of latency

and 2, the proposed shortest path estimation approach: MFWA shows better link load and latency performance as compared to the existing shortest path estimation approaches (FWA, DA [21], and EDA [22]). Generally, the cache load in forward and backward path is different, due to traffic of interest and data flow. The dynamic change in the

load is easily maintained and shortest path with respect to cache load is easily identified by using MFWA algorithm by eliminating the invalid paths utilizing as iterative matrix. The graphical depiction of link load and latency performance is shown in the Figs. 2 and 3.

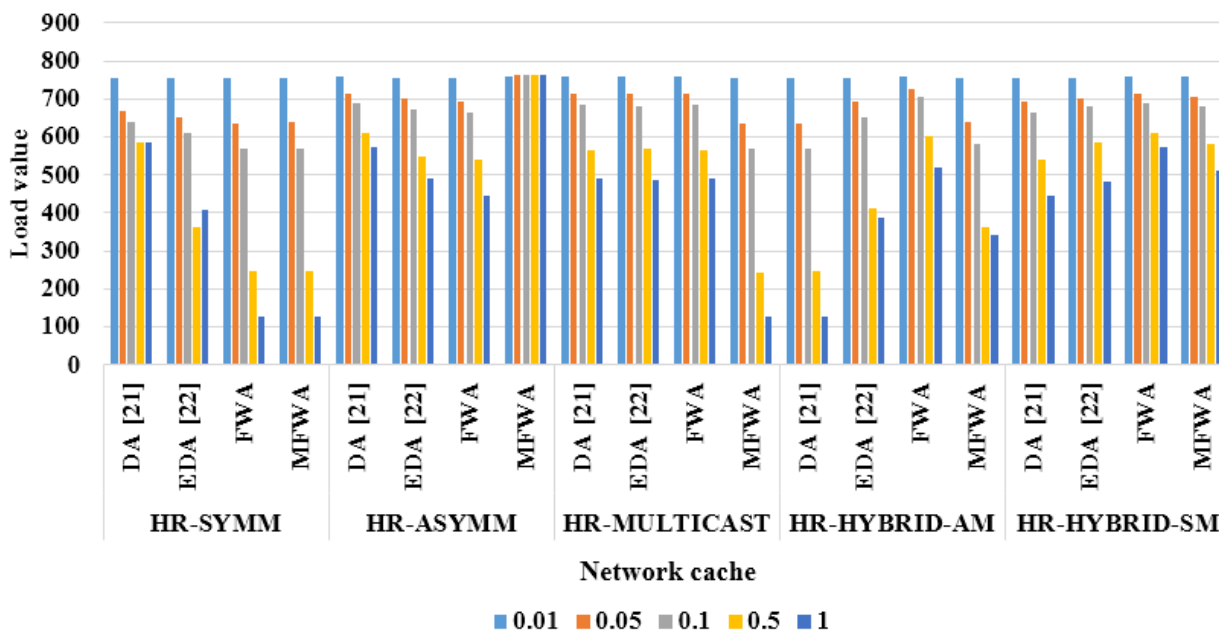


Figure.3 Graphical comparison of proposed and existing methodology by means of link load

Table 3. Quantitative analysis of proposed and existing work in terms of cache hit ratio

		Cache hit ratio (%)				
Cache policies	Shortest path estimation algorithm	Network cache				
		0.01	0.05	0.1	0.5	1
HR-SYMM	DA [21]	0.74	11.8	15.9	23.1	23
	EDA [22]	0.87	12.36	18.12	27.91	34
	FWA	0.94	16.3	25.4	67.7	83
	MFWA	0.91	17.9	27.90	70	86.78
HR-ASYMM	DA [21]	0.91	16.3	25.3	67.8	83
	EDA [22]	0.93	9.72	25.91	45.02	71.26
	FWA	0.98	8.5	12.5	29.2	41.6
	MFWA	0.91	17.9	27.90	70	86.78
HR-MULTICAST	DA [21]	0.9	16.3	25.4	67.7	83.4
	EDA [22]	0.9	15.09	20.85	53.97	78
	FWA	0.32	6.2	9.9	26.1	35
	MFWA	0.9	16.3	25.3	67.9	83
HR-HYBRID-AM	DA [21]	0.29	6.5	10.2	26.8	37
	EDA [22]	0.22	5.96	19.75	42.75	41
	FWA	0.17	4.6	7.4	21	31.7
	MFWA	0.89	16	23.9	52.4	55
HR-HYBRID-SM	DA [21]	0.98	8.5	12.5	29.2	41.62
	EDA [22]	0.43	6	11.02	24.7	37.86
	FWA	0.497	6.5	9.6	19.9	24.6
	MFWA	0.5	7.1	10.5	23.8	32.5

Correspondingly, Table 3 shows the performance of proposed and existing short path estimation methods in light of cache hit ratio. In this scenario, the experimental investigation is validated with five dissimilar network caches (0.01, 0.05, 0.1, 0.5, and 1). The validation outcome shows that the proposed shortest path estimation approach: MFWA gives

better outcome in all cache management policies; HR-SYMM, HR-ASYMM, HR-MULTICAST, HR-HYBRID-AM, and HR-HYBRID-SM compared to the existing shortest path estimation algorithms (FWA DA [21], and EDA [22]). The graphical depiction of cache hit ratio is shown in the Fig. 4.

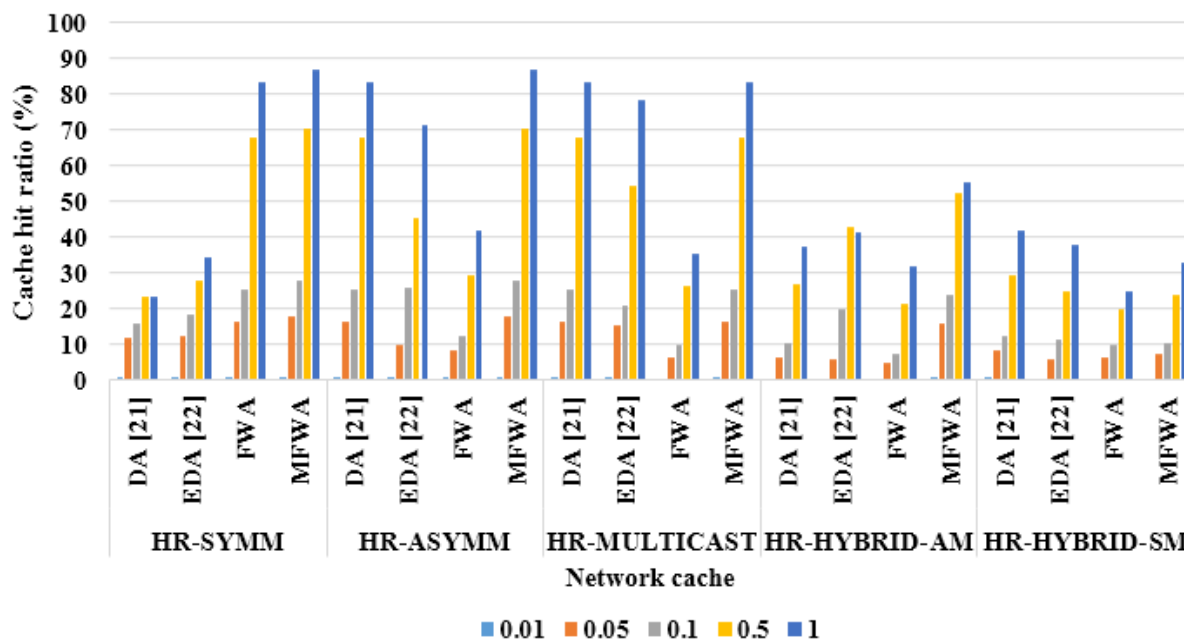


Figure.4 Graphical comparison of proposed and existing work by means of cache hit ratio

5. Conclusion

In this research article, an automatic cache management architecture is developed that dynamically reassigns information to the caches of an ICA. The reassignment decisions are based on real time data such as, locality of request and observed popularity. The main goal of this experiment is to develop a proper shortest path estimation algorithm. Here, MFWA is proposed to identify the shortest path in every subnet and between subnets that helps to recover the cache memory data quickly. Compared to the existing shortest path estimation algorithms (FWA, DA and EDA), the proposed shortest path estimation algorithm delivered an effective performance by means of quantitative analysis and comparative analysis. From the experimental study, the proposed shortest path estimation algorithm showed better result in cache management by means of cache hit ratio, link load, and latency. The experimental result showed that the proposed methodology improved cache hit ratio up to 4-30% as related to the existing methods. In future work, a new hybrid shortest path estimation algorithm is developed for further improving the performance of cache management in ICN.

References

- [1] S. Saha, A. Lukyanenko, and A. Ylä-Jääski, "Efficient cache availability management in information-centric networks", *Computer Networks*, Vol.84, pp.32-45, 2015.
- [2] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, "Distributed cache management in information-centric networks", *IEEE Transactions on Network and Service Management*, Vol.10, No.3, pp.286-299, 2013.
- [3] J. Duan, X. Wang, S. Wang, and S. Xu, "Cache Management for Information-centric Networking", In: *Proc. of IEEE 17th International Conference on Computational Science and Engineering*, pp. 1927-1933, 2014.
- [4] Y. Yamamoto, J. Takemasa, Y. Koizumi, and T. Hasegawa, "Analysis on Caching Large Content for Information-Centric Networking", In: *Proc. of the ACM Conference on Information-Centric Networking*, 2018.
- [5] B. Banerjee, A. Kulkarni, and A. Seetharam, "Greedy Caching: An optimized content placement strategy for information-centric networks", *Computer Networks*, Vol.140, pp.78-91, 2018.
- [6] F. Al-Turjman, "Fog-based caching in software-defined information-centric networks", *Computers & Electrical Engineering*, Vol.69, pp.54-67, 2018.
- [7] K. Sato, T. Kamimoto, R. Shinohara, and H. Shigeno, "Cache management with extended interest for information-centric networking", In: *Proc. of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, pp.245-250, 2016.
- [8] R. Huo, R. Xie, H. Zhang, T. Huang, and Y. Liu, "What to cache: differentiated caching resource allocation and management in information-

- centric networking”, *China Communications*, Vol.13, No.12, pp.261-276, 2016.
- [9] H. Huang, L. Rui, W. Zheng, D. Niu, and X. Qiu, “CMLP: Exploiting caches at multiple levels of proxies to enhance seamless mobility support in information-centric networks”, *China Communications*, Vol.13, No.10, pp.86-107, 2016.
- [10] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li, “Design and evaluation of probabilistic caching in information-centric networking”, *IEEE Access*, Vol.6, pp.32754-32768, 2018.
- [11] J. Khan, C. Westphal, J. Garcia-Luna-Aceves, and Y. Ghamri-Doudane, “NICE: Network-oriented Information-centric Centrality for Efficiency in Cache Management”, In: *Proc. of ACM Conference on Information-Centric Networking (ACM-ICN 2018)*, 2018.
- [12] G. Carofiglio, L. Mekinda, and L. Muscariello, “Joint forwarding and caching with latency awareness in information-centric networking”, *Computer Networks*, Vol.110, pp.133-153, 2016.
- [13] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, “An optimal cache management framework for information-centric networks with network coding”, In: *Proc. of IFIP networking conference*, pp.1-9, 2014.
- [14] H. Zhang, S. Zhu, R. Xie, T. Huang, and Y. Liu, “Intelligent resources management system design in ICNing”, *China Communications*, Vol.14, No.8, pp.105-123, 2017.
- [15] C. Li, S. Gong, X. Wang, L. Wang, Q. Jiang, and K. Okamura, “Secure and Efficient Content Distribution in Crowdsourced Vehicular Content-Centric Networking”, *IEEE Access*, Vol.6, pp.5727-5739, 2018.
- [16] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, “A minimum cost cache management framework for information-centric networks with network coding”, *Computer Networks*, Vol.110, pp.1-17, 2016.
- [17] S. Paul, B. Banerjee, A. Mukherjee, and M. K. Naskar, “Priority-based content processing with Q-routing in information-centric networking (ICN)”, *Photonic Network Communications*, Vol.32, No.3, pp.348-358, 2016.
- [18] M. Bilal, and S.G. Kang, “A cache management scheme for efficient content eviction and replication in cache networks”, *IEEE Access*, Vol.5, pp.1692-1701, 2017.
- [19] J. Shi, X. Wang, and M. Huang, “ICN-based cache-aware routing scheme in MSN”, *Ad Hoc Networks*, Vol.75, pp.106-118, 2018.
- [20] I. Psaras, W.K. Chai, and G. Pavlou, “In-network cache management and resource allocation for information-centric networks”, *IEEE Transactions on Parallel and Distributed Systems*, Vol.25, No.11, pp.2920-2931, 2014.
- [21] B. Banerjee, A. Seetharam, A. Mukherjee, and M.K. Naskar, “Characteristic time routing in information centric networks”, *Computer Networks*, Vol.113, pp.148-158, 2017.
- [22] S. Ahmed, R.F. Ibrahim, and H.A. Hefny, “Mobile-based Routes Network Analysis for Emergency Response Using an Enhanced Dijkstra’s Algorithm and AHP”, *International Journal of Intelligent System and Engineering*, Vol.11, No.6, pp.252-260, 2018.
- [23] S. Hougardy, “The Floyd–Warshall algorithm on graphs with negative cycles”, *Information Processing Letters*, Vol.110, No.8-9, pp.279-281, 2010.