



Maximum Concurrent Flow Solutions for Improved Routing in IoT Future Networks

Abou-Bakr Djaker¹ · Bouabdellah Kechar¹ · Hossam Afifi² · Hassine Moun gla^{2,3}

Received: 25 November 2021 / Accepted: 9 November 2022 / Published online: 22 November 2022
© King Fahd University of Petroleum & Minerals 2022

Abstract

Due to the advent of IoT and the increasing interest of billions of Internet users towards video contents, a huge multimedia flows has been generated, and as a consequence, a massive load is applied on the underlying core network. This change can affect the network stability and lead to potential performance degradation (such as congestion and delays). This is because multimedia flows are hungry in bandwidth, and also the classical routing protocols currently used in IoT core network (like OSPF) are not adapted yet to support the heavy and the large scale of multimedia traffics with a good quality of service (QoS). In this paper, we introduce the multicommodity-based routing by proposing two contributions, Maximum Concurrent Flow Protocol (MCFPr) and Cache-based Maximum Concurrent Flow (C-MCF). They are conceived based on the Maximum Concurrent Flow approach in order to optimize the routing of multimedia data in the backbone of IoT networks. Both contributions are studied and compared with the state-of-art approaches under different scenarios showing good results, especially in the number of data packets sent (improvement by 50%), and in the transmission time (50% faster compared to the majority), which makes them promising solutions for a rapid and efficient routing in IoT core networks.

Keywords Internet of things · Maximum concurrent flow problem · Routing · Caching · Centrality · Named data networking

1 Introduction

The fast development of IoT and data transmission protocols and technologies [1,2] has improved the human life in various fields (e.g., smart homes and cities, smart transportation, smart agriculture, e-health, and even for Hajj & Umrah [3]). Recently, the IoT has known an increasing interest towards multimedia contents (especially videos) through

many emerging applications and services (such as smart IPTVs, Webinars, Video social media) that results in unprecedented huge multimedia traffics. This change brings many challenges as well, particularly in the bandwidth management and data routing [4], and can impact the performance in the IoT backbone network (in terms of congestion, delays, packets loss, etc.). This is because the multimedia services are high bandwidth-consuming, and the current IGP routing protocols (such as OSPF and RIP) cannot handle this enormous quantity of flow because they are originally designed for the Traditional Internet [5]. Since all IGP routing protocols use *Ford–Fulkerson* algorithm to find the maximum flow, only one flow (i.e., single commodity) is routed through the shortest-path even when there are a lot of data flows, they will be routed one by one. In this case, the bandwidth of the shortest path is not fully leveraged, and it takes a lot of time to send all the flows.

In order to cope with the all aforementioned challenges, a promising approach called MCFP (Maximum Concurrent Flow Problem) [6] is used by our (two) contributions that are designed for the IoT core network (TCP/IP architecture): (i) MCFPr (Maximum Concurrent Flow Protocol) that is similar to OSPF but with optimized bandwidth management that

✉ Abou-Bakr Djaker
djaker.aboubakr@edu.univ-oran1.dz

Bouabdellah Kechar
kechar.bouabdellah@univ-oran1.dz

Hossam Afifi
hossam.afifi@telecom-sudparis.eu

Hassine Moun gla
hassine.moun gla@parisdescartes.fr;
hassine.moun gla@telecom-sudparis.eu

¹ Laboratoire RIIR, Département Informatique, Université Oran1 Ahmed Ben Bella, Oran, Algérie

² UMR 5157, CNRS, Institut Polytechnique de Paris, Télécom SudParis Saclay, Paris, France

³ Université de Paris, LIPADE, F-75006 Paris, France



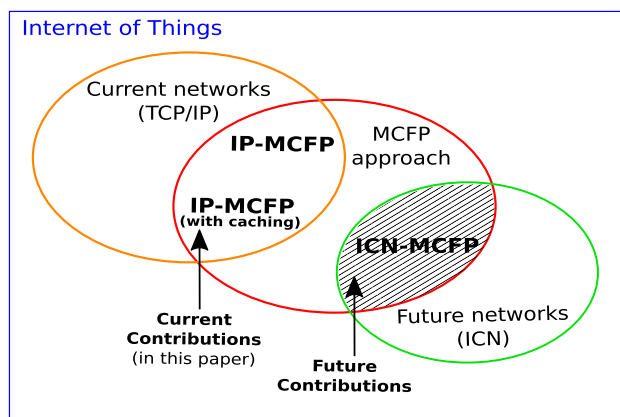


Fig. 1 The contributions of this paper

allows sending many flows simultaneously, and (ii) C-MCF (Cache-based MCF) which is the improvement of the previous solution that has integrated the cache memory in order to increase the data availability and reduce the latency time, C-MCF is inspired from the ICN (Information-Centric Networks) [7]. Both contributions present a lot of advantages for the routing in IoT backbone (e.g., less data packets to send, short transmission time, optimal link utilization). Figure 1 gives a comprehensive illustration, the left intersection between the TCP/IP network and MCFP approach shows our two contributions presented in this paper. We will discuss about our future contributions (the right intersection) as perspectives later in Sect. 8.

The rest of this paper is organized as follows: Sect. 2 overviews the existed IP routing protocols, and presents two major problems about flow maximization (i.e., MFP and MCFP) that are the basis of this paper. In Sect. 3, we introduce research works that are inspired by OSPF, the widely deployed routing protocol, and present also a recent state-of-the-art of MCFP. After that, we present the motivations behind the use of MCF approach as an original routing technique in IoT network in Sect. 4. We continue in the same section by explaining the scenario of the concurrent flows and its benefit for IoT. The contribution of this paper is twofold, we devote Sect. 5 to present and evaluate our first routing solution based on MCFP for the IoT networks, whereas the second contribution given in Sect. 7 is designed to improve the routing much more by integrating the mechanism of content caching within the nodes. The paper is then concluded in Sect. 8 followed by future perspectives.

2 Background

There are many research works in the literature that focus on improving the quality of data routing in the IoT network

(especially in WSN [8–10]). In this article, we target the routing in IoT core network.

In this section, we present briefly the existed routing protocols designed for IP networks, then we discuss their constraints face to the evolving content requests. After that, we tackle two interesting approaches for computing the highest flow supported in the graph (i.e., Maximum Flow and Maximum Concurrent Flow Problems).

2.1 Routing Protocols Overview

For the reason of scalability and processing performance, the Internet is composed of two types of network (intra- and inter-domains), each type has a specific routing protocol: IGP (Interior Gateway Protocol) designed for Autonomous Systems (ASs), and EGP (Exterior Gateway Protocol) used to link between ASs. Moreover, the IGP has two sub-categories: Distance Vector Routing (DVR) and Link-State based Routing (LSR), and both of them have their proper strategy to disseminate the routing information. The LSR is more privileged for large and complex networks than DVR because of many advantages [11]. The OSPF (Open Shortest-Path First) is a LSR protocol, it undergone a series of improvement [12,13], and it is the most efficient routing protocol deployed on the Internet for many decades. The contributions presented in this paper are inspired by this protocol.

2.2 Routing Protocols Constraints

The IP networks are knowing a continuous increase in the number of Internet users. This increase is behind the proliferation of different kind of devices (such as Smartphones, Tablets, Smart TVs), and it is accompanied with special interests in video contents. During the last decades, the Internet users have become interested in the multimedia and video streaming more than other services, such as Video-on-Demand (VoD) in IPTVs, live video conferencing, social media live videos and so on. These multimedia applications are demanding in the scarce network resources (i.e., Bandwidth), and depending on their quality definition, the bandwidth consumption will vary. According to Cisco annual report [14], in 2023 over 82% of the Internet traffic will be video flows. This upcoming change makes the routing protocols in challenge to find efficient solutions for better bandwidth management alongside with satisfied QoS. In this regard, we built our both contributions (Sects. 5 and 7) on a promising paradigm called 'Maximum concurrent flow' that is presented in the following subsection.

2.3 Routing Models

There are many and different routing protocols in the literature, some of them are based on the number of hops,

while the others consider the metric of cost, such as OSPF. This protocol is basically the distributed version of the well-known Dijkstra algorithm used to construct a loop-free SPT (Shortest-Path Tree) at each router. This SPT supports a maximum flow, and in order to calculate it, there is a multitude of algorithms. Particularly, we are interested in the following two approaches:

2.3.1 Maximum Flow Problem (MFP)

It uses the algorithm of Ford–Fulkerson to find the augmenting paths that allow increasing the flow. When there is no possible augmented path, the maximum value of flow is achieved, and the algorithm terminates. It is essential to remind that this method is applied in the case of a single commodity issued from a single source to a single destination. This type has been improved and extended to support a multitude of heterogeneous traffic as shown next.

2.3.2 Maximum Concurrent Flow Problem (MCFP)

This type belongs to the multicommodity flow problems, it was studied extensively in the literature [15–21]. Its particularity is instead of sending one commodity as in MFP, it issues multiple commodities together along the shortest paths subject to the available capacity. More specifically, a portion γ for each commodity are sent together at each iteration. γ is calculated using the size of commodity (i.e., the flow) and the capacity of the link in the shortest path [6]. Since MCFP is also based on augmented SPT, the final value of γ is the minimum among all the links in the SPT [22]. Hereafter, we give its ILP (Integer Linear Programming) formulation [6]:

$$\begin{aligned} & \text{maximize } \gamma \\ & \text{Subject to:} \end{aligned} \tag{1}$$

$$\sum_{v \in V} f_i(u, v) - \sum_{v \in V} f_i(v, u) = 0, \quad \forall u \in V, \quad \forall i \tag{2}$$

$$\sum_{i=1}^k f_i(u, v) \leq c(u, v), \quad \forall (u, v) \in E \tag{3}$$

The objective of MCFP also is to maximize the value of concurrent flow γ (Eq. 1). Since we are in the context of multicommodity, each commodity must have the same value of γ . Also, two conditions must be verified: (i) flow conservation constraint (Eq. 2), where the total ingress flow to a node must equal the egress, and (ii) capacity constraint (Eq. 3), the total flow issued must not exceed the capacity of the link. Here, $f_i(u, v)$ represents the commodity i sent from node u to node v , and $c(u, v)$ is the capacity of the link between them.

3 State-of-the-Art

In this section, we outline some related works of OSPF protocol since it is the basis of our contribution. Then, we give a brief state-of-the-art about the MCFP approach and the proposed algorithms (in Sect. 3.2). We also discuss about some research works for OSPF and MCFP in the context of future Internet architecture NDN (Named Data Networking) [23]. NDN is an emerging paradigm that has gained a wide attention; recently, it is conceived mainly for better contents routing and management.

3.1 OSPF (Open Shortest-Path First)

Due to the efficiency of OSPF routing and its wide applicability for many years, several works have been based on it. They developed specific versions for the current Internet architecture (based on TCP/IP protocol stack), and for the future architecture based on the ICN (Information-centric Networking) paradigm [7]. We can cite, for example, some of them as shown in Fig. 2: (i) MOSPF (Multicast OSPF) [24], an extension of OSPF to support multicast routing, (ii) GOSPF (Green OSPF) [25], an enhanced protocol designed to reduce the energy consumption by adapting the network topology to the data traffic, (iii) OSPFN (OSPF for NDN) [26], an enhanced routing protocol designed for the next-generation of the Internet (i.e., NDN) that uses name prefixes (similar to URL structure) instead of IP addresses.

3.2 MCFP (Maximum Concurrent Flow Problem)

There are many optimization problems under the umbrella of multicommodity (as shown in Fig. 3), each problem has a specific objective whether to maximize the flow (e.g., *max. MCNF*), or maximize concurrent flows (e.g., *MCFP*) or even minimize the cost (e.g., *min. cost MCNF*). These models are specially used to resolve the network design and routing problems in many disciplines. As example, Fig. 4 shows some application fields of MCFP [27,28]. Although the concept of multicommodity was proposed in the 1970s, it was not used due to the complex and time-consuming resolution algorithms existed at that time [19]. However, there are many research works, particularly MCFP algorithms [18–20,27] that were published recently. In general, an MCFP algorithm can belong either to the rerouting or the incre-

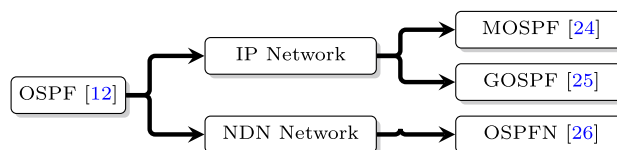


Fig. 2 OSPF-based protocols

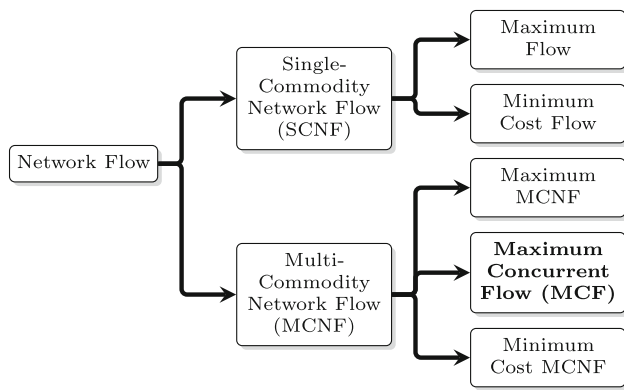


Fig. 3 Network Flow optimization problems

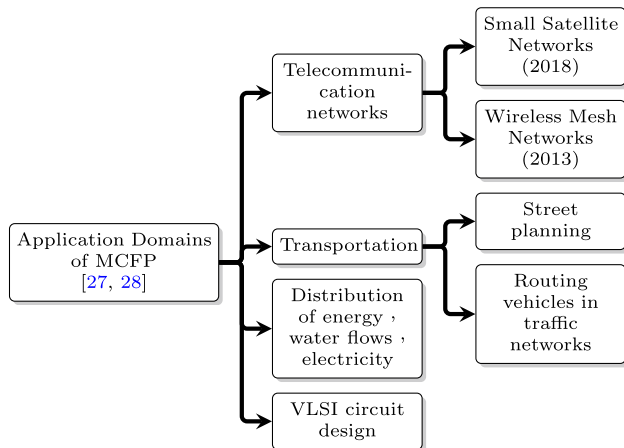


Fig. 4 Examples of MCFP application domains

menting approach. We present hereafter some of them (see Fig. 5):

- **Rerouting Algorithms**, the principle of this category is to assign each commodity to the best shortest path. If its size exceeds the capacity of the path, the overflowed quantity will be rerouted to another best shortest path in order to verify the MCFP conditions. The first algorithm proposed in this class specifically and in MCFP in general is the algorithm of Shahroukhi and Matula [6] that were the first who presented the concept of concurrent flow. The major idea behind is the use of a mathematical function called '*length function*' associated with each link. This function represents the cost (i.e., the degree of congestion) of the path, it decides which one will receive the rerouted flow. The algorithm [6] is the foundation of many other methods such as Klein et al. algorithm [33] that improves their running time; Goldberg [34] algorithm introduces the random choice of commodities at the opposite of Radzik [32] that follows a deterministic round-robin steps.

- **Incrementing Algorithms**, as the name indicate, the flow is incremented iteratively along the shortest paths according to specific conditions. The idea of Young [31] was to push small flows progressively. This method took much time to finish. It was then improved in Garg and Könemann algorithm [15] that increases the flow until saturating the smallest capacity within the path through one iteration, this change gave better performance. After that, many algorithms have followed its principle, firstly Fleischer [16] that applies approximation steps instead of the exact approach; and Karakostas [30] that optimizes the construction of shortest paths for the commodities sharing the same source node by calling the *Dijkstra* algorithm only once. Madry [29] also proposed a special data structure to output rapidly the shortest paths using a special dynamic-graph method called '*Decremental All-Pair Shortest Path Problem*' (DAPSP), once the paths are found, the MCF method will be applied.

Recently, an interesting work was published by Zhang et al. [19] that applies the MCFP as a method to optimize the data routing in NDN future networks. This work will be tackled in the next subsection.

3.3 MCFP in Future Internet Architecture

To the best of our knowledge, the work [19] is the first and the only one that resolves the routing problem in NDN networks using the MCFP approach. The authors present an optimization model for the content traffic, and resolve the violated MCFP constraint (i.e., flow conservation) caused by the NDN smart forwarding feature. The key solution is to consider the smallest edge-capacity in the constructed SPT (Shortest-Paths Tree) as the maximum capacity. Note that the SPT is the topology that carries the concurrent flows. Based on this model and inspired by the work of Fleischer [16], they proposed also two MCF algorithms:

- *FRR (improved Fleischer and Random Rounding)*, it has the same length function as in paper [16]. When the iterations of flow incrementation terminate, the cumulative flow will be scaled down to verify the MCFP capacity condition. The particularity of this algorithm is the adaptation of the smart forwarding plane to select randomly the outgoing faces when there are many paths. A '*probability*' field was hence added to the *FIB (Forwarding Information Base)* table,
- *MCMP (Multicommodity Multipath)*, the length function in this time is different from paper [16]. It helps finding the shortest path for each commodity. The residual capacity of the selected path will decrease according to the size of commodity, this will update the length function

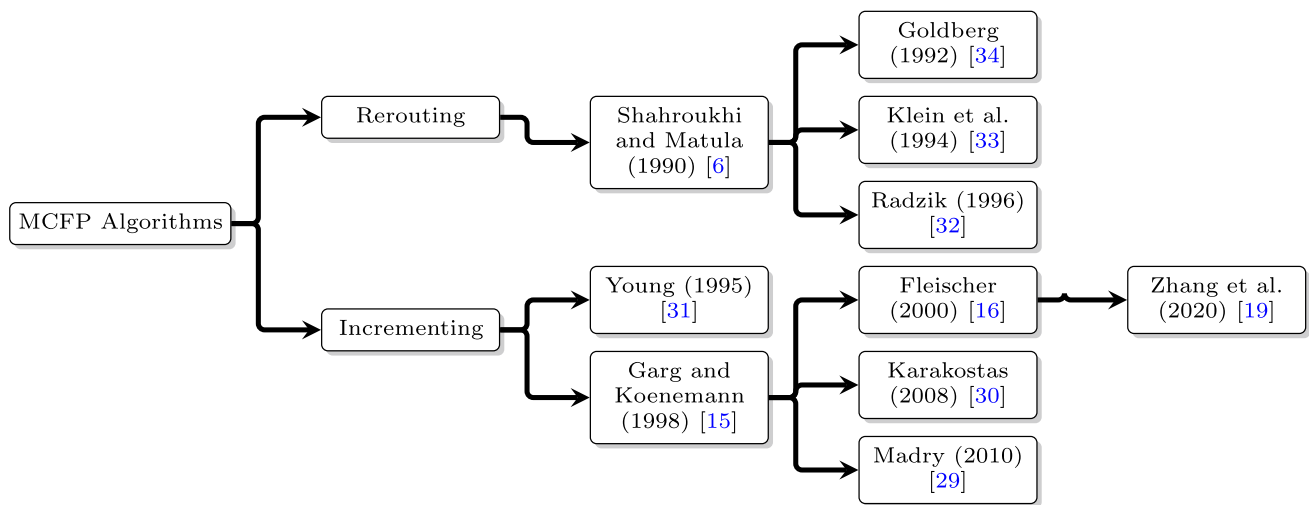


Fig. 5 Proposed improvements to MCFP algorithms

as well. If there are saturated links, they will be avoided in the next iteration. This process is repeated until the destination nodes become unreachable. The present algorithm does not need to decrease the final flow at the end because the constraints are already proved.

Similarly, we propose MCFP solutions to improve the routing in IoT (IP networks). The motivations behind this work are presented in the next section.

4 Motivations of Applying MCF in IoT Networks

The IoT encompasses a variety of applications with different scenarios, each application has specific requirements. Basically, the IoT is divided into two categories [35,36]: Scalar and Multimedia IoT (M-IoT). Obviously, these two different networks process different kind of data and they require specific capabilities. Figure 6 shows a comparison between the characteristics of IoT data in both networks. The multimedia use-cases are hungry applications in terms of network resources mainly in bandwidth consumption, therefore, they need to be optimized. According to Cisco’s report [14], the daily traffic in the Internet will exceed 2.6 exabytes in 2023 and over 82% of the global traffic will be multimedia. This represents the big load that will be applied on the network. In addition to this, there are many emerging applications in M-IoT (e.g., see Fig. 7) that will increase further the multimedia traffic. Another reason that has motivated this work is the impact of Covid-19 pandemic on the Internet utilization. The social distancing and the lockdown have motivated the use of many multimedia and video streaming services [37] such as the videoconferencing (used for remote working, vir-

Types and characteristics of IoT data		
	Scalar data	Multimedia data
Data types	simple	heterogeneous
Processing cost	low	high
Storage space	minimal	big
Bandwidth utilization	minimal	greedy
Energy consumption	low	high

Fig. 6 Comparison between IoT data requirements

tual classrooms, etc.) and for entertaining also (like Netflix and YouTube that have been classified as the top applications used during this period [37]). As a consequence, it is mandatory to cope with these heterogeneous and heavy traffics on the underlying network. It is necessary then to find efficient mechanisms to optimize the circulation of multimedia data.

Since there are a multitude of multimedia applications in M-IoT, we chose the use-case of ‘Multimedia Delivery Networks’ [38] (Fig. 7) because the video traffics in these networks are very dense and heterogeneous. In this regard, we use the technique of concurrent flows (MCFP) to exploit efficiently the network links (i.e., bandwidth). We can then model this scenario following the multicommodity approach as shown in Fig. 8. The multicommodity flows (in the aforementioned figure) are the heterogeneous multimedia data

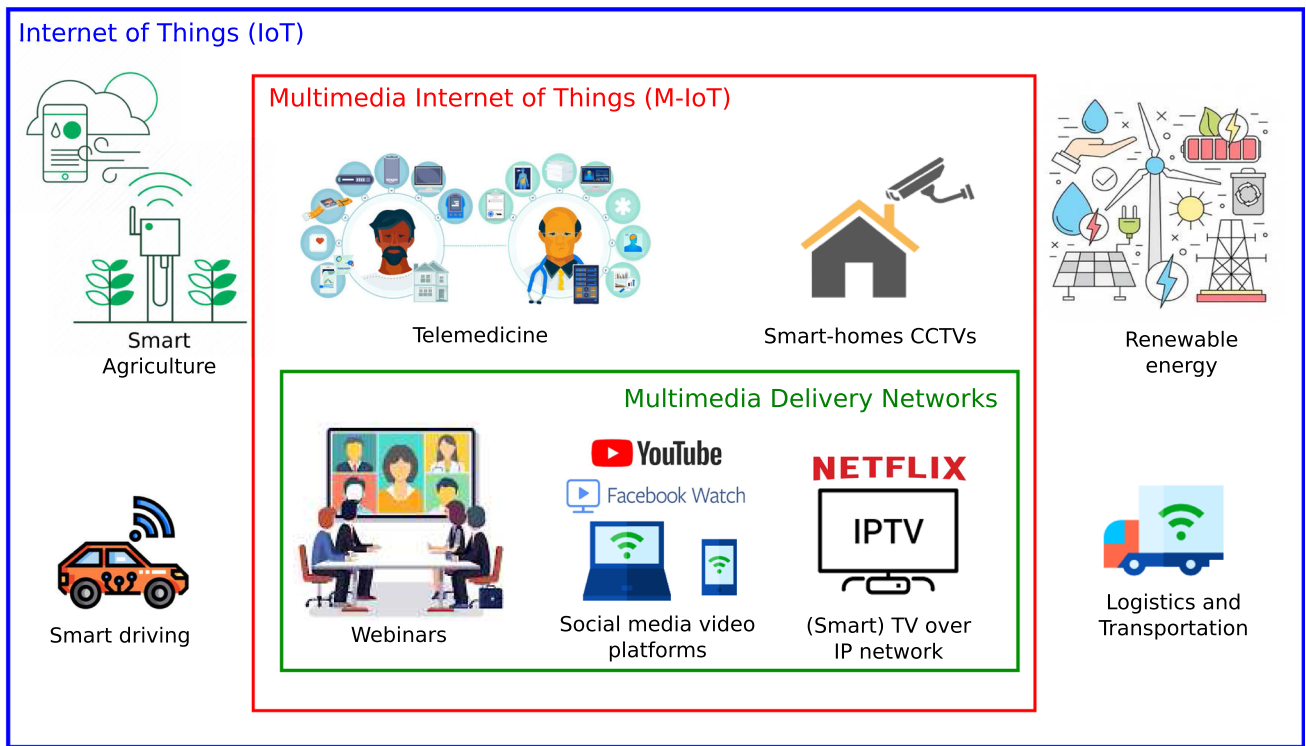


Fig. 7 Some applications in multimedia IoT

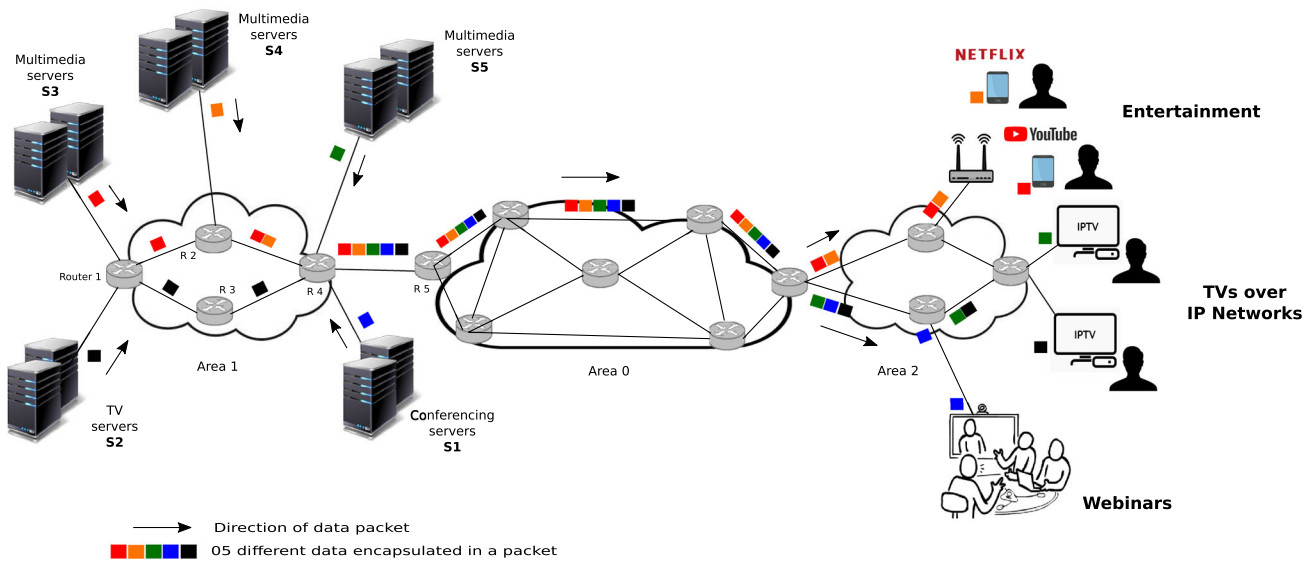


Fig. 8 The application of MCF in the Multimedia Delivery Networks

transmitted from the sources (i.e., the servers) towards the destinations (i.e., the Internet users). Figure 8 shows a simple example about the deployment of the multicommodity MCF technique in the Multimedia Delivery Networks. We suppose that the users have already made their requests and wait for the responses that are represented in the same figure. We suppose also that the multimedia data are stored in the multimedia servers. The concurrent flows or more specifically

the data packet (that encapsulates many chunks of different commodities) will be forwarded over the shortest path. As example, in Fig. 8 the router R4 sends to the router R5 the data packet that have encapsulated the multimedia contents requested by the users. Upon receiving this packet by a router, it decides whether to split it into many packets or just forward it to the next router. At the end, each user will receive the content requested. This mechanism has many advantages, it can

reduce the overall transmission time by using less shortest paths for sending lots of data, and in the same time preventing congestion (a feature in MCFP). As a result, this idea represents an interesting solution for the routing in M-IoT for the current and long term.

Based on MCF approach, we propose two contributions in this paper:

- Maximum Concurrent Flow Protocol (MCFPr) in Sect. 5
- Cache-based Maximum Concurrent Flow (C-MCF) in Sect. 7

5 Maximum Concurrent Flow Protocol (MCFPr)

It have been proved in paper [39] that the MCFP can be adapted to IoT to optimize the data routing through a centralized approach. In this section, we present our decentralized approach (our first contribution), it is a routing protocol based on MCFP model designed for the backbone (area 0) of IoT networks.

5.1 MCFPr Description

MCFPr is an IGP (Interior Gateway Protocol) conceived based on OSPFv3 protocol [13]. In the topology discovery phase, all routers advertise their link-state information in order to have the same *LSDB (Link State Database)*. The permanent exchange of *Hello* packets between the routers helps to verify the accessibility of neighbor nodes. If a router did not interact in this process after many attempts, it will be considered as dead. As a consequence, it will be removed from all LSDBs using an update LSA message. When all routers maintain the same LSDB, they will apply the routing algorithm to find the shortest paths. MCFPr differs from OSPFv3 in the algorithm applied, OSPF constructs the SPT using the *Dijkstra* algorithm while MCFPr uses the MCFP algorithm[22]. Many shortest paths will be then selected in the routing table for the concurrent transmissions. This concurrency is modeled by a packet that encapsulates many data chunks (as shown in Fig. 9), each chunk is associated to a different commodity. The maximum size of the chunk is determined by the same algorithm[22]. The IP and TCP headers are also adapted by adding the corresponding IP addresses and their ports. It is worth mentioning that the maximum length of IPv6 packet can be up to 65,535 bytes, and the fragmentation is performed when the size of packet exceeds the MTU (Maximum Transmission Unit) of the path [13].

Subsequently, it is necessary to highlight the strengths of MCFPr: (i) Robustness: when a link fails, the traffic is impacted only in that part since the data are sent by chunks

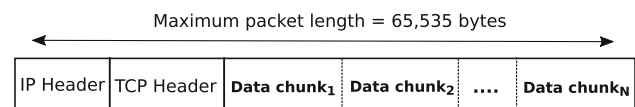


Fig. 9 The structure of MCFPr data packet

over many different shortest paths, and they are not lost completely. In other words, there is no single-link failure. As a consequence, the routers will adapt to this change by choosing another shortest path, (ii) Network resource management: the conception of this protocol guarantees an optimal utilization of the available shortest paths and their capacities. Also, it prevents the congestion and wasting bandwidth thanks to MCFP flow control, (iii) Flexibility and efficiency: MCFPr is built based on IPv6 that offers many assets in the long term, specially for IoT networks. Moreover, this protocol can be applied in any IoT use-case, especially in the applications eager for bandwidth (e.g., video streaming services) thanks to its optimal routing strategy.

5.2 Performance Evaluation

We used Jupyter Notebook software [40] to implement our MCF solution, and *NetworkX* library [41] to generate random topologies. This library provides many useful functions to simulate complex networks such as IoT. Since we are interested in the core network, the evaluations are applied on random topologies with different scales varying from 100 to 300 nodes. The network size within this interval is considered as the standard for the Internet backbone according to Cisco report [42] and other works [43,44]. In the evaluation process, we consider the following metrics: (i) Delay and transmission times, both of them are necessary to determine the fastest approach in the data transmission; (ii) Links utilization, we count the total number of the links utilized; (iii) Capacity utilization, it represents the total capacity utilized (i.e., the overall bandwidth use) in the network; (iv) The number of data packets sent; and (v) the Maximum flow (MF). Since the data traffic is highly dependent to the topology used, the MF metric was added to know the maximum flow supported by each approach. Some approaches optimize and reduce the initial topology, this leads to different values of MF. All the aforementioned metrics are among the essential measures to assess the performance in IoT core network. They give informative insights about each method and more importantly, their behaviors in different conditions.

We compare our contribution with the state-of-the-art approaches that are presented in table 1. The motivation of this choice is threefold:

1. These methods are designed to improve the performance of the data traffic,

Table 1 A comparative table of the evaluated approaches

	OSPF	HPAC	SDP	ACO	MCF
Objective	Finding the shortest paths	Reducing transmission cost	Reducing delay time	Finding the optimal shortest paths	Optimizing bandwidth consumption in the shortest paths, and maximizing the throughput
Topology made	Lowest cost SPT	Lowest cost SPT	Least delay SPT	Least hops SPT	Maximum flow SPT
Number of commodities/ transmission	Single commodity	Single commodity	Single commodity	Single commodity	Multicommodity
Main strengths	Shortest path computation	Data migration and Gomory–Hu transformation	Least-delay path computation	Probabilistic shortest path computation	Concurrent transmission of heterogeneous data, maximizing the flow and preventing congestion
Domain of application	Distance related problems	QoS and QoE content delivery	Delay sensitive applications	Distance related problems	Wide applications of goods shipment in different fields, network planning and routing optimization

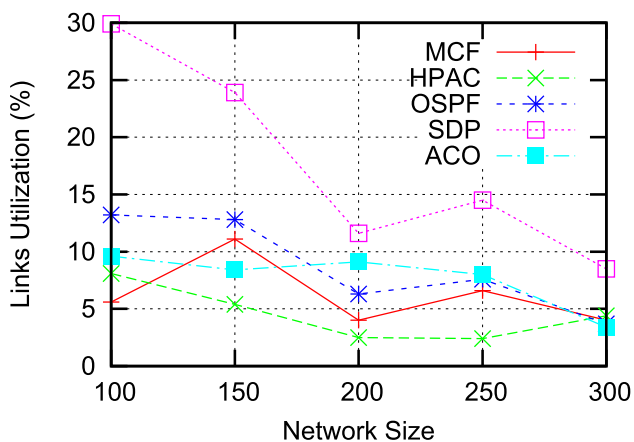
2. These methods are implemented and assessed extensively in the literature, showing good results in different optimization areas such as finding the shortest paths, maximizing data flow, and reducing the time cost,
3. These high-performance and single-commodity methods should be compared with the multicommodity MCF solution.

As a consequence, they are presented as follows: (i) *OSPF*, the well-known and the widely used routing protocol until nowadays, it uses the *Dijkstra* Algorithm to find the shortest paths; (ii) *HPAC* [45], the idea of this method is to migrate highly requested data from its original source to specific nodes near the Internet users in order to serve their future requests; (iii) *SDP (Shortest Delay Path)* routing, this is a basic version of algorithm [46] that constructs routing paths based on delay time and the network load; (iv) *ACO (Ant Colony Optimization)* routing, it is an elementary approach of the work [47] in which the shortest paths are discovered by Ants (as agents). The principle is simple: each Ant while searching the destination (i.e., the source of food), it releases a chemical substance called 'pheromone.' This latter will vaporize with time unless another Ant passes by this path and releases its proper pheromone. When the Ants finish the exploration phase, we find the shortest path (i.e., the path used by all Ants to reach the destination). It is the one that has much quantity of cumulative pheromone. Table 1 presents more details about the aforementioned approaches.

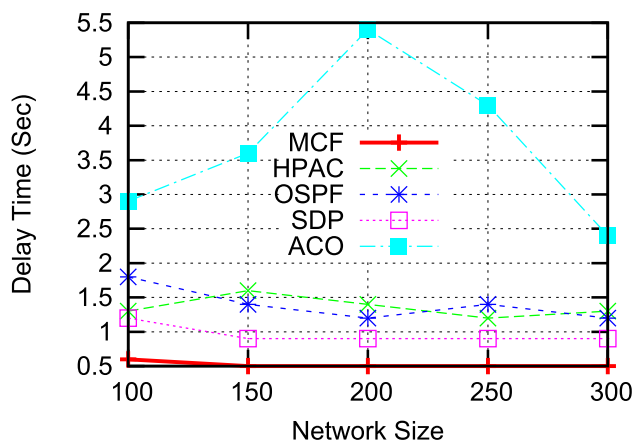
We performed two evaluations based on the network size and the number of commodities. In each one, we measured a set of metrics for all approaches. The description of each evaluation and the interpretation of the results are given hereafter:

5.2.1 The First Evaluation (Fig. 10)

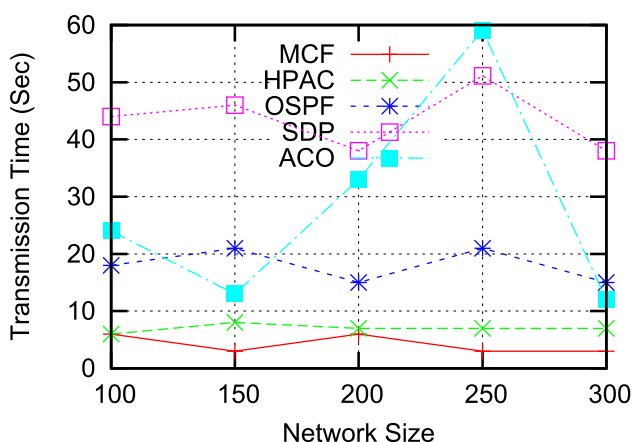
We set the number of commodities to three, and we vary the number of nodes to see the behavior of the approaches. Initially, in Fig. 10a MCF has a low percentage of link utilization compared to the majority, this is due to the flow concurrency feature. Instead of using a path for sending each data packet independently (i.e., single commodity transmission), MCF can encapsulate and send them together through common shortest paths unlike HPAC that uses one shortest path for each transmission. But, HPAC reduces the network topology by applying the *Gomory–Hu (G-H)* transformation, this leads to a small set of links. Also, HPAC has a data migration feature that shortens the transmission distance. Since MCF can deliver encapsulated data packets of many commodities, the needed shortest paths are systematically less than the usual transmission, this leads as a consequence to reduce the delay and the transmission time (Fig. 10b and c, respectively) which results finally in a better time-cost compared to the other methods. In the next Fig. (10d), the maximum flow is calculated according to *Miller and Naor* [48] that initially adjust any input topology with multiple-source multiple-destination to a single-source single-destination. The maximum flow procedure is then applied on the resulted topology. We notice that OSPF, SDP, and ACO have the same values, this is because they did not apply any reduction or modification to the network topology. At the opposite, MCF generates an augmenting *Shortest-Paths Tree (SPT)*, and HPAC uses the *G-H* method to derive a small tree from the initial topology. MCF has medium results in this measure. For the overall bandwidth consumption (Fig. 10e), MCF has a high rate, which is obvious because it is designed to exploit optimally the network resources, and in the same time preventing the



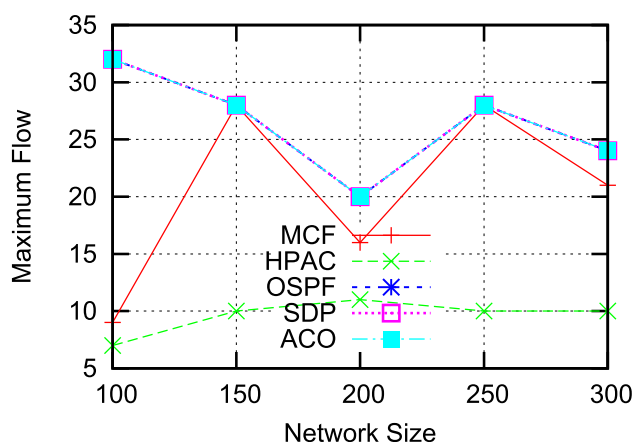
(a) Links Utilization



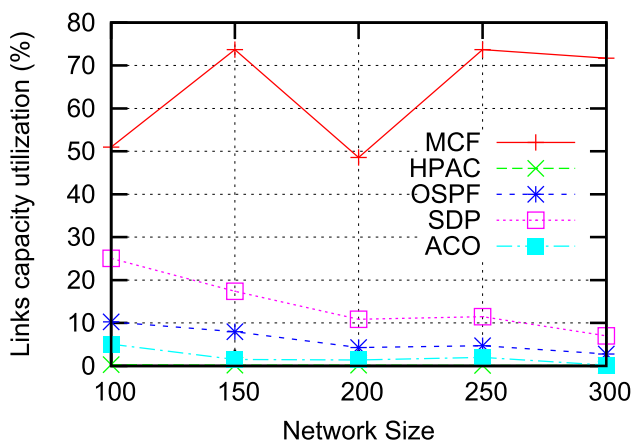
(b) Delay



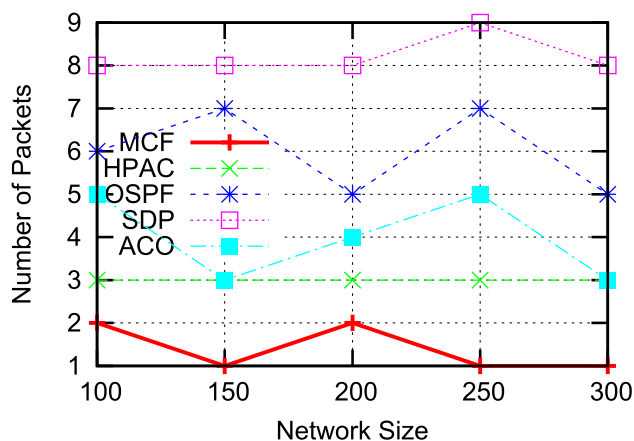
(c) Transmission time



(d) Maximum Flow



(e) Capacity utilization



(f) Number of packets

Fig. 10 Evaluations in case of 03 commodities sent over different network scales

congestion [49]. Lastly but not the least, the number of data packets sent over the network (Fig. 10f) is also important to judge the routing performance. MCF has few data packets sent and it outperforms the other methods thanks to the encapsulation feature.

5.2.2 The Second Evaluation (Fig. 11)

We chose random topologies of 300 nodes, and we varied the number of commodities transmitted in the network. This assessment aims to test the robustness of the routing approaches against the increasing data traffic. Starting by Fig. 11a, we see that MCF has a very low link utilization compared to HPAC and the other routing approaches. This is due to the augmenting SPT that maintains the best paths to all destinations. Also, the concurrency feature of MCF represents its main asset, it selects the common shortest paths to issue different data packets to different destinations. This advantage leads as a consequence to reduce the delay and the transmission time as shown in Fig. 11b and c. Since MCF and HPAC derive their specific topologies (i.e., augmented SPT and G-H tree, respectively) from the initial topology, the available paths were obviously decreased, only the best ones are kept. Hence, the values of maximum flow in both trees are decreased as well, at the opposite to the other approaches. Again, we apply here the method of *Miller and Naor* [48] to find the maximum flow similar as done previously in the first evaluation (Fig. 10d). Although the increase of the data flow in the network, the MCF routing adapts to this change. It updates its SPT by selecting the efficient shortest paths with sufficient bandwidth; we can see the amelioration of MCF in Fig. 11e. Since MCF encapsulates various data in common packets and use common shortest paths to send them, the total number of the data packets sent is small thanks to the features of the multicommodity MCF (i.e., maximizing the transmission throughput and preventing the congestion). The corresponding measure is shown in Fig. 11f. We recall that the chunks (within the data packet) have the same size, it is inevitable then that the increase in commodities will impact the value of this size. Figure 12 illustrates this scenario by providing the variation of MCF ratio over the increasing commodities.

Our MCF solution can be improved further. We recall that it is based on an end-to-end transmission (i.e., the data are fetched from the data servers, and sent back to the Internet users that have initially made their requests). If many users request the same data in different times, their redundant requests will reach the data servers and the response will be sent back multiple times. In this communication, there is a waste of the network resources, particularly in the bandwidth consumption. Thus, it is necessary to optimize our solution by taking into account the aforementioned scenario.

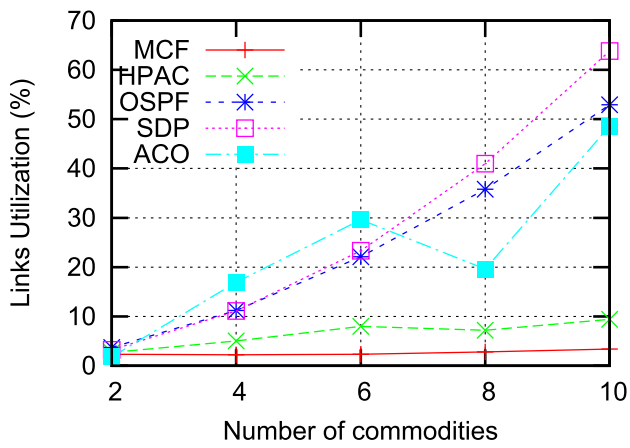
In the next section, we introduce briefly the concept of Information-Centric Networking 'ICN' (i.e., the next generation of the Internet) and its caching property. Our goal is to take advantage of the ICN-caching principle to improve further the MCF routing in IoT. Thus, the most requested data (specifically, the multimedia) will be cached near the Internet users. These caching nodes will serve data rapidly and efficiently to the future requests better than accessing the distant data servers. Our second contribution is then presented in Sect. 7.

6 The Caching Feature and its Advantages for IoT

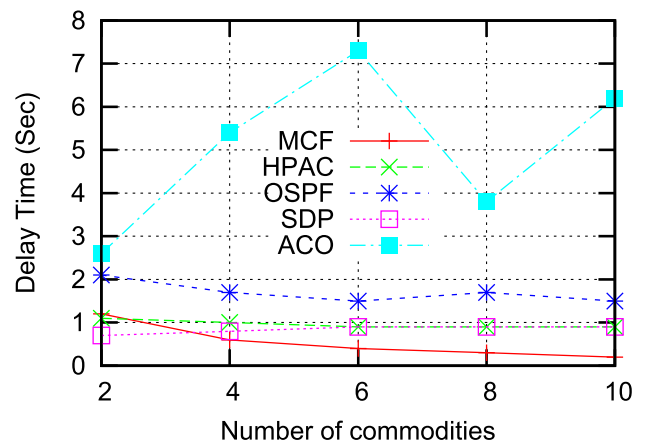
Due to the increasing demands on the multimedia contents recently by the Internet users, in parallel with the proliferation of multimedia gadgets and smartphones, the current Internet architecture (TCP/IP) will become exhausted. It will need to be reshaping to support the next generation of massive data traffics. A considerable effort has been made two decades ago by many researchers to think for solutions. They conceived a new vision of networking called 'Information-Centric Networking' [7]. The ICN is a communication paradigm designed specifically to handle the contents more efficiently than the TCP/IP model. One of the main strengths of the ICN is the replication of contents (i.e., caching) in the intermediate nodes of the network. This technique is useful to respond to the future requests of different users in a short and optimized manner. This feature hence brings several advantages in terms of network performance such as optimizing the traffic flow, reducing the latency and transmission time, and also increasing the availability of contents. These qualities are required in the Multimedia IoT. Similarly, we integrate the principle of caching in our MCF-based routing for IP networks. This solution inherits the ICN-caching properties and its aforementioned advantages. For this reason, our contribution is presented by analogy to NDN model (Named Data Networking [23], the most successful ICN architecture) in the next section.

7 C-MCF: Cache-based Maximum Concurrent Flow Model

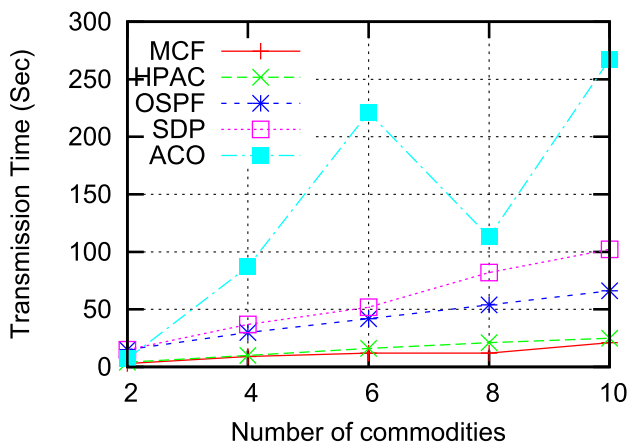
The caching feature plays a vital role in ICN/NDN networks. There are many works in the literature that address this concept. Particularly, the cache placement and replacement strategies are widely emphasized [50,51]. They are based on an important factor called "*the popularity of content*" that is measured by the number of Interests (i.e., requests) sent. The NDN routers with caching capabilities react efficiently in the network to fulfill the upcoming Interests. They respond



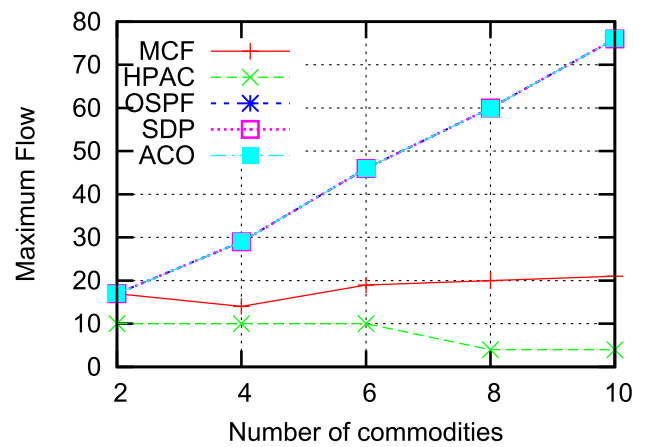
(a) Link utilization



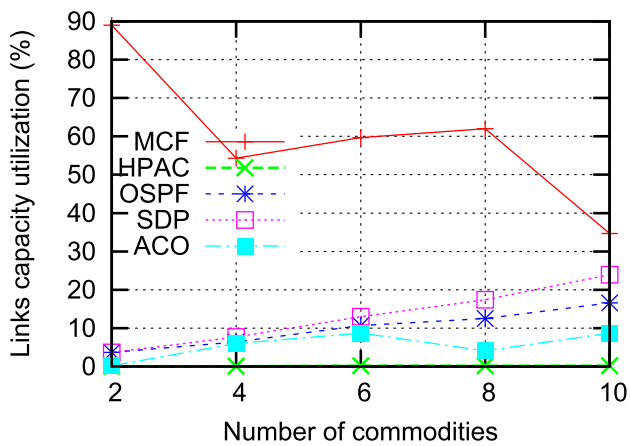
(b) Delay



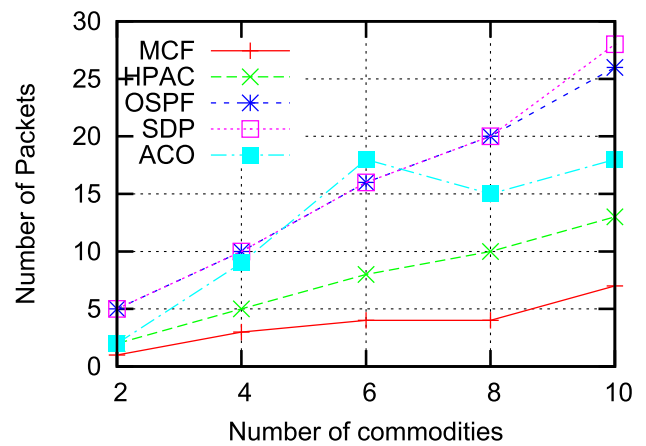
(c) Transmission time



(d) Maximum Flow



(e) Capacity utilization



(f) Number of packets

Fig. 11 Evaluations in case of different commodities sent over a network of 300 nodes

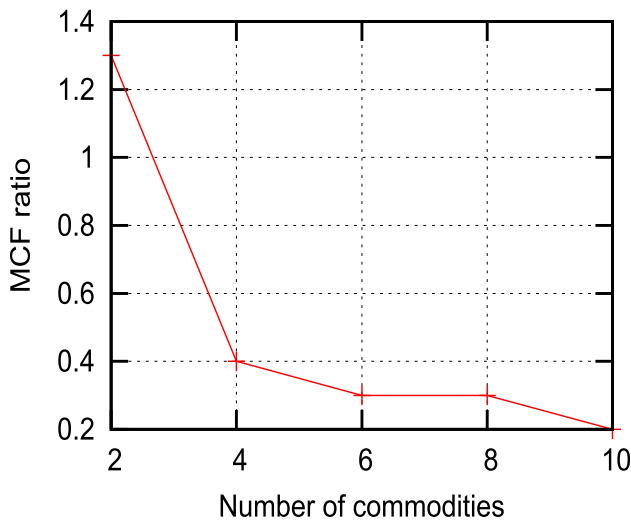


Fig. 12 MCF ratio with commodities variation over network of 300 nodes

with the appropriate data stored in their cache memory. This feature have many advantages in NDN networks especially for the data routing.

In our (second) contribution, we propose a routing solution (for IP networks) based on MCF method, and we integrate the caching functionality as utilized in ICN/NDN networks. Some network nodes are selected as caching nodes according to specific criteria (described later). These nodes account the most demanded data during a period of time Δt , and save a copy of them. Upon receiving the requests, they respond with the corresponding data through concurrent flows. In other words, the caching node encapsulates the responses of the requests received in common data packets, and send them

back to the users following the shortest paths. This scenario is illustrated in Fig. 13. Again, we consider the use-case of 'Multimedia Delivery Networks,' and we suppose also that many requests have been made before in order to select the popular contents eligible for caching. The routers R12 and R13 are the caching nodes that apply the MCF procedure to send the data packets. Hereafter, we present the mathematical model that resolves both the caching nodes placement and the MCF problems:

7.1 Model Formulation

Formally, our model is built up of two objective functions:

- Part I: it minimizes the distance between the group of clients c and the node that caches the highly-requested data d . The new cache placement will be in node s' . Hereafter, the first objective function (Eq. 4):

$$s'(d) = minimize \sum_i x_{n,i} * P_{\Delta t}(I_n^d) \tag{4}$$

$$Subject\ to: P_{\Delta t}(I_n^d) = 1$$

$$n \in N, 3 \leq i \leq c, \text{ and } c \subseteq C$$

The parameters and constraints of Eq. (4) are described in Table 2.

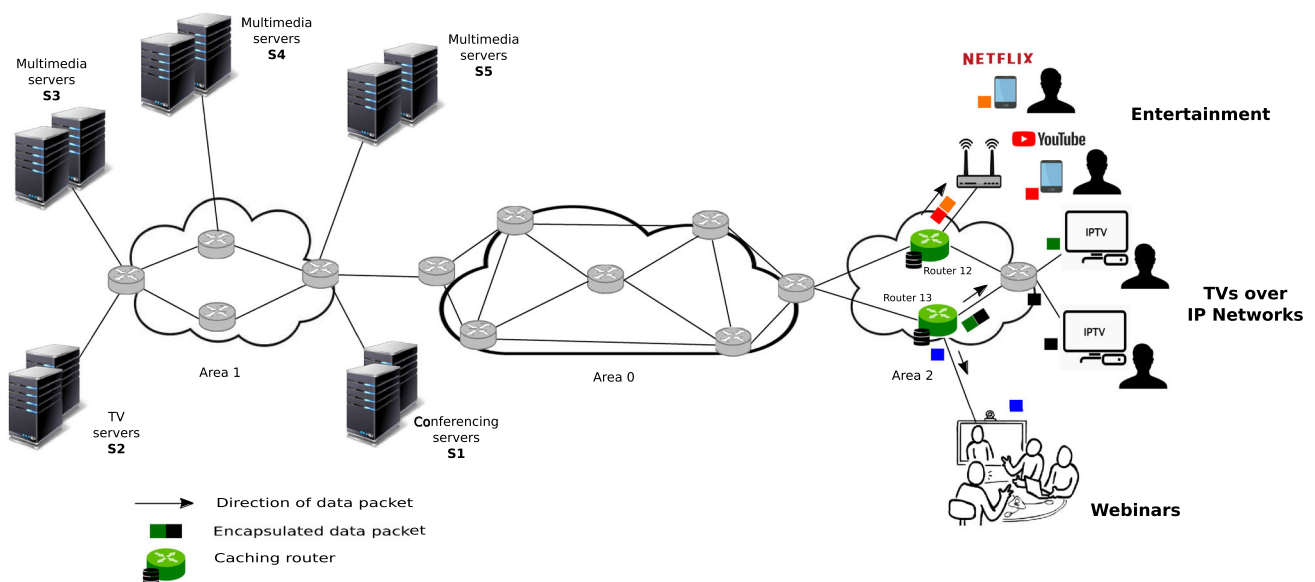


Fig. 13 The application of the MCF and the caching techniques in M-IoT networks

Table 2 Model parameters and constraints

Parameter notation	Designation
N	The set of nodes
d	The popular data
i	A client belongs to a group of clients c
C	The set of all clients
$I_{i,n}$	The request (Interest) sent by client i and received by node n
$I_{i,n}^d$	Request for data d
$P_{\Delta t}(I_n^d)$	The popularity of requesting data d during a period of time Δt , calculated by node n
$x_{n,i}$	The distance between the cache of data d (hold by node n) and the client i
$x_{a,b}$	$x_{a,b} = 1$ means one hop from node a to b
$x_{s',c}$	Sub-SPT between source node s' and group of clients c . It is derived from the global topology
$s'(d)$	Optimal placement selected for caching the data d
$B(x_{s',c})$	Bandwidth of the path between the node s' and client i
γ	Maximum concurrent throughput, it is the maximum size allocated for each data chunk
$D_{s'}$	The set of data requested (popular and not) stored in the cache of node s' . $D_{s'} = \{d, \bar{d}\}$
$D_{s',c}^{a,b}$	The set of data $\{d, \bar{d}\}$ to transmit from node a to b in the tree $x_{s',c}$
$f_{s',c}^{Max}$	The maximum flow between the source node s' and the group of clients c
Decision variables	
$P_{\Delta t}(I_n^d)$	$P_{\Delta t}(I_n^d) = 1$ if $(\frac{\sum_i x_{s',c}}{\sum_i I_{i,n}} > 0.5)$, 0 otherwise
Constraints	
$3 \leq i \leq c$, and $c \subseteq C$	It should be three clients at least in order to calculate the popularity of requests.
$\sum f_{a,b} - \sum f_{b,a}$	Flow conservation constraint
$f_{a,b} \leq B(x_{a,b})$	Capacity constraint

- Part II: once the new data source (cache node) s' is determined, it becomes the root of a sub-SPT (Shortest-Paths Tree). In this tree, the leaves are the destination nodes (i.e., group of clients c) that receive the data sent from the caching node. As a consequence, the scale of the initial topology is reduced. This leads to numerous advantages (e.g., shortest transmission time). Subsequently, the MCF approach will be applied to the new generated tree. The second objective function is presented as follows (Eq. 5):

$$\text{maximize } \gamma \tag{5}$$

$$\text{Where : } \gamma = \min \left\{ \frac{B(x_{a,b})}{\sum_{s',c} D_{s',c}^{a,b}} \text{ for all } a, b \in x_{s',c} \text{ and } x_{a,b} = 1 \right\}$$

Subject to:

$$\sum f_{a,b} - \sum f_{b,a} = \begin{cases} \gamma * D_{s'} & \text{if } a = s', a \neq b \\ 0 & \text{if } a, b \neq s' \\ -\gamma * D_{s'} & \text{if } a = i, a \neq b \\ a, b \in x_{s',c} \end{cases}$$

$$f_{s',c}^{Max} = \min \{ B(x_{s',c}), 3 \leq i \leq c \}$$

$$f_{a,b} \leq B(x_{a,b}), a, b \in x_{s',c}$$

The parameters and constraints of Eq. (5) are described in Table 2.

It is clear that our model belongs to the multicriteria optimization problem. According to the methods of combining maximization with minimization functions, and in order to uniform the objectives, the second function will be adapted to the minimization model by passing through the LP (Linear Programming) transformation (see Eq. 6):

$$\text{maximize } \gamma \longrightarrow \text{minimize } \frac{1}{\gamma} \tag{6}$$

Finally, we form our bi-objective function presented in Eq. 7:

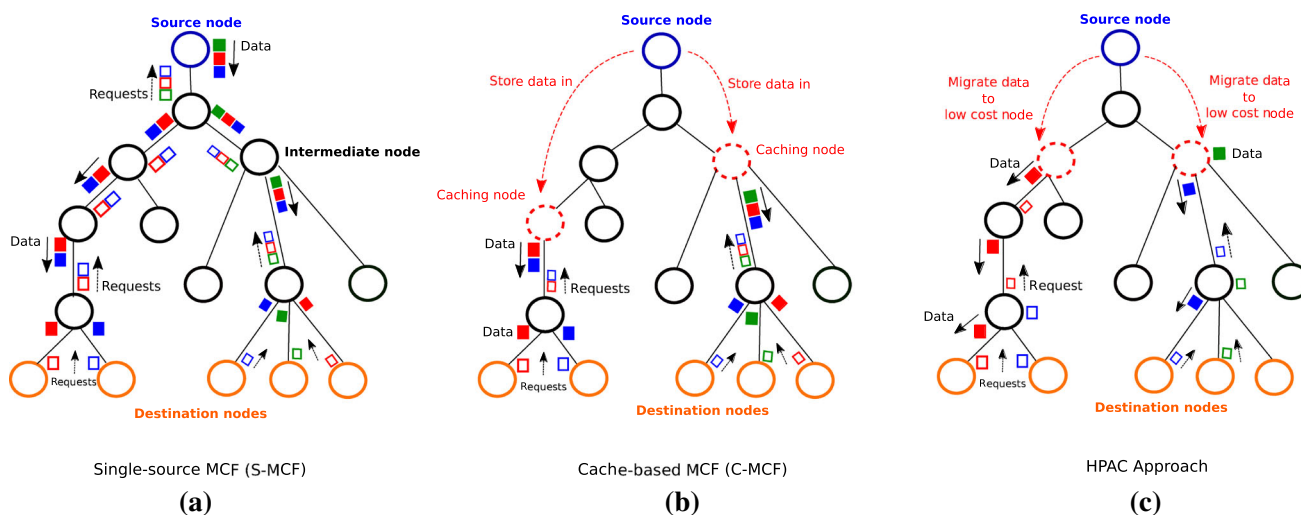


Fig. 14 S-MCF, C-MCF and HPAC schemes

$$minimize\left(\frac{\sum_i x_{n,i} * P_{\Delta t}(I_n^d)}{\gamma}\right) \tag{7}$$

The description of each parameter (used in the previous equations) is presented in Table 2.

7.2 C-MCF Algorithm

Based on the previous model, we propose our MCF algorithm (see algorithm 1) with complexity $O(m^2.n)$. Its particularity compared to the other algorithms is in the following points: (i) the application of MCFP to optimize the routing of data in IoT IP networks, and (ii) the use of caching feature to respond efficiently to the requests. C-MCF combines between the advantages of the multicommodity concurrent flows and the caching in intermediate nodes. Instead of applying the MCF procedure in the data-source node, it is executed in a distributed way by the caching nodes. Thus, the cache placement is crucial, for this reason we choose this placement according to two factors: (i) Request popularity $P_{\Delta t}(I_n^d)$, it helps the intermediate nodes to know the most requested content by the consumers during a period of time, and (ii) Closeness Centrality measure $C_c(n)$, it detects the central nodes of the network (i.e., the 'influenceable' nodes that belong to the majority of the shortest paths). The nodes with highest Closeness Centrality value have great impact in the dissemination of contents within the network. Once the caching nodes are set, each one constructs a sub-SPT to reach the nearest group of users. If a shortest path is used in a tree, it will not be included in the others, in order to prevent the congestion.

Algorithm 1 Cache-based MCF (C-MCF)

- Input:**
- A graph $G = (N, E)$ where N : Nodes and E : Edges, $|N| = n$ and $|E| = m$;
 - Commodities information (s_i, t_i, d_i) : source node s_i , destination node t_i , size of the data requested d_i .
- Output:**
- Maximum concurrent throughput γ_{Max}
- Main:**
- 1: Compute the requests popularity $P_{\Delta t}(I_n^d)$ at each node ;
 - 2: Compute the Closeness Centrality $C_c(n)$ associated to each node n ;
 - 3: Select the node u with highest value of product of $P_{\Delta t}(I_n^d)$ and $C_c(n)$ as a caching node ;
 - 4: Construct the Shortest-Paths Tree (SPT) from node u to the nearest destinations ;
 - 5: Find Maximum concurrent Throughput γ_{Max} of SPT.

7.3 Performance Evaluation

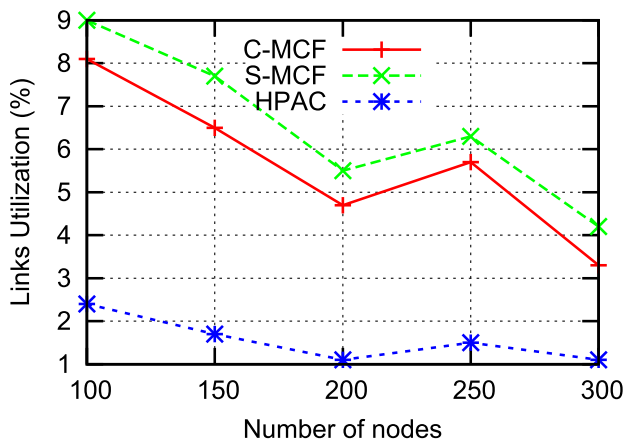
In this part, we evaluate and compare our contribution (C-MCF) with other methods: S-MCF (Single-source MCF, the approach used by our first contribution in Sect. 5 that use MCF without caching), and HPAC [45]. The reason why we choose these methods is because they are designed to optimize the data traffic efficiently, and they showed better performance (as shown previously in Figs. 10 and 11). We also want to examine the efficiency of the caching feature added to our contribution, and compare this solution with other methods within the same scenarios. S-MCF does not have any caching node, and the data are stored only in the source node. In contrast, C-MCF can deploy many nodes to store a copy of the highly requested data (see Fig. 14). Both of them apply the principle of the multicommodity MCF (i.e., encapsulated packet with various inner data). C-MCF is also compared with HPAC that use caching with lowest

Table 3 Inputs and assumptions for the evaluations

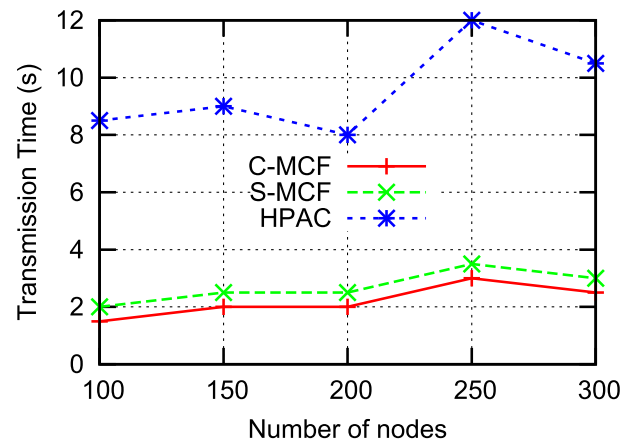
	Random graph(s)	Random consumers	Size of requested data
Experiment 1 (Fig. 15)	05 graphs (from 100 to 300 nodes)	05 consumers	10 MB
Experiment 2 (Fig. 16)	01 graph (300 nodes)	05 cases (from 2 to 10 consumers)	10 MB
Experiment 3 (Fig. 17)	05 graphs (from 100 to 300 nodes)	05 consumers	$10 \text{ MB} \leq \text{size} \leq 40 \text{ MB}$
Experiment 4 (Fig. 18)	01 graph (300 nodes)	05 cases (from 2 to 10 consumers)	$10 \text{ MB} \leq \text{size} \leq 40 \text{ MB}$

cost. However, this solution is designed for single commodity flows. Figure 14 depicts the circulation of Requests and Data packets (border-colored and full-colored squares, respectively) according to each approach. In S-MCF and C-MCF, the data packet can contain one to more data chunks (i.e., full-colored squares), and they must have the same size. If any user requests a data that exceeds this size, it must be divided into chunks and encapsulated into the data packets.

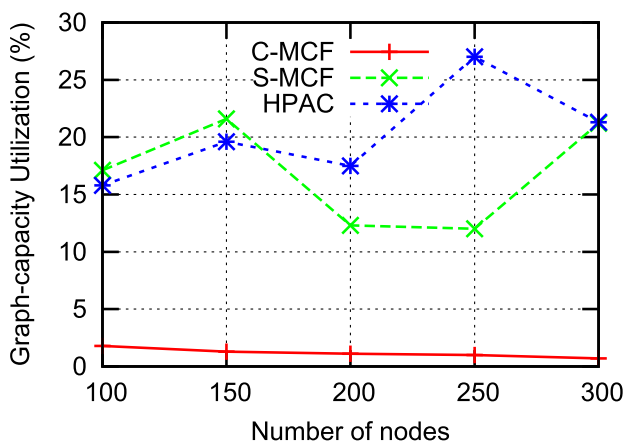
We used Jupyter Notebook software [40] to implement these approaches, and the NetworkX library [41] to generate random graphs in each experiment as shown in Table 3. Also, the choice of users, and their requests are generated randomly. Initially, We calculate the utilization percentage for the links and their capacities (i.e., the total number of links utilized, and the overall capacity exploited), then we measure the transmission time. The concurrent throughput is



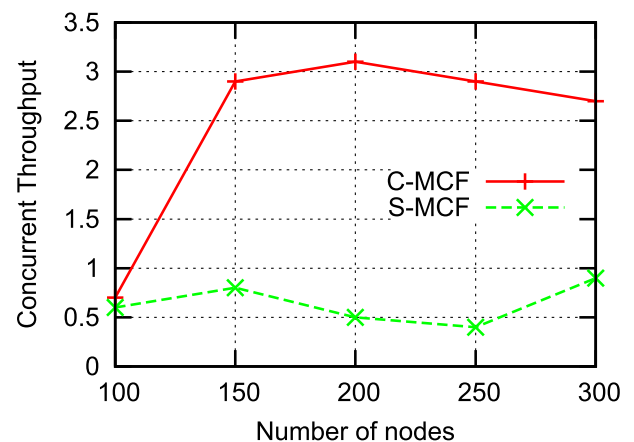
(a) Links Utilization



(b) Transmission Time



(c) Capacity Utilization



(d) Concurrent Throughput

Fig. 15 Evaluation of the approaches (Experiment 1)

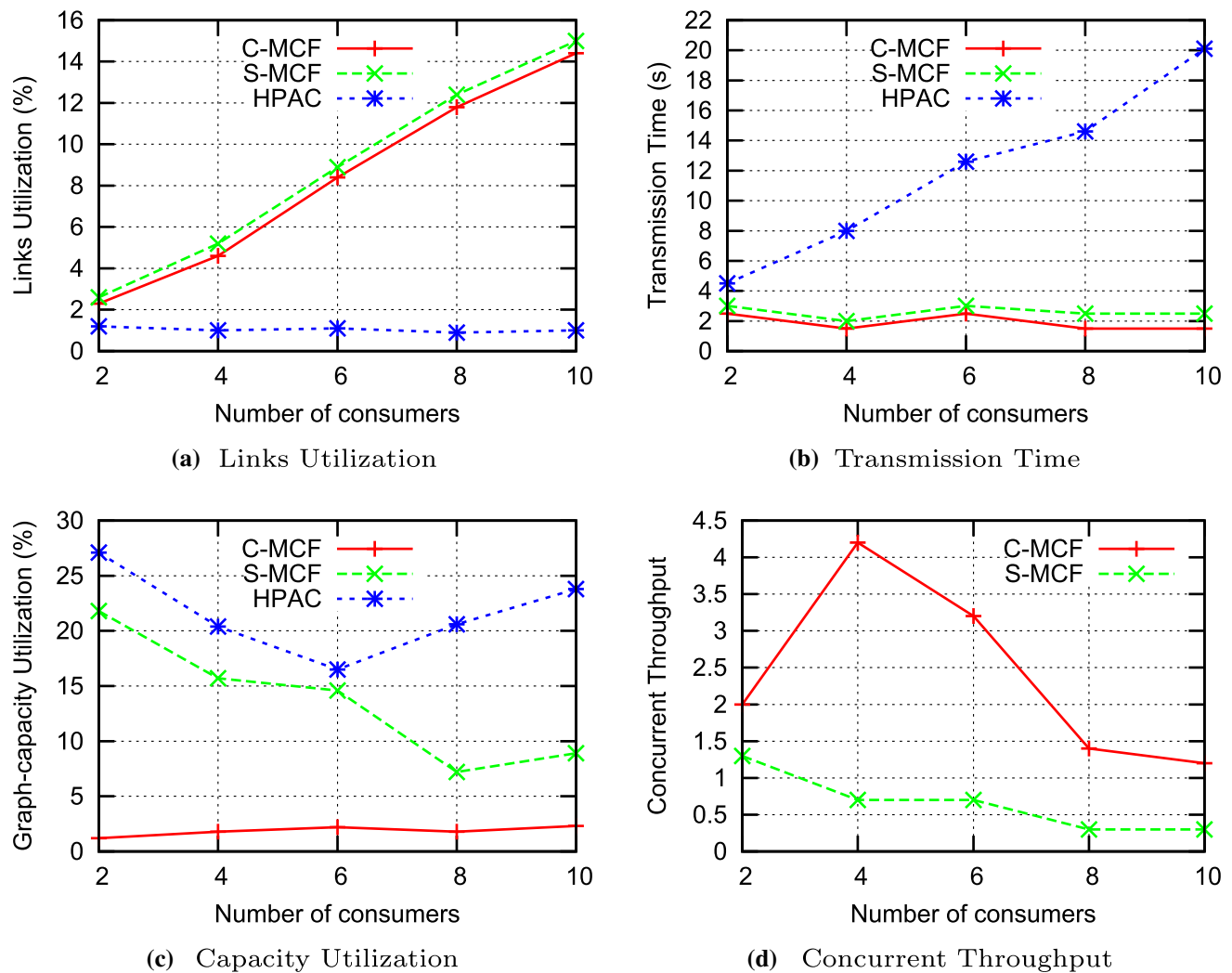


Fig. 16 Evaluation of the approaches (Experiment 2)

also calculated but only in S-MCF and C-MCF since HPAC does not support the concurrency in the transmission.

In the first experiment (Fig. 15), we consider 05 random consumers for each graph varying from 100 to 300 nodes, they made requests for many contents (e.g., YouTube videos) with a size of 10 MB. Similarly for the third experiment (Fig. 17), but in this case, the size of the requested content is variable (i.e., between 10 MB and 40 MB). The reason why we chose these data sizes is because the Internet shorter videos are generally the most interesting and viewed by the Internet users [52,53]. These evaluations are chosen to see the behavior of the approaches in random graphs that scale up, and especially the impact of the content size on the network performance. In the other experiments (Figs. 16 and 18), we fixed the scale of the graph to 300 nodes, and we vary the number of the potential consumers from 2 to 10 clients. Again, in these two cases, we test the performance with a fixed and variable data size. The aim of experiments 2 and 4 is to exam-

ine our contribution (C-MCF) in a complex network that has an increasing number of clients with variable Interests.

The results of all experiments (from Figs. 15 to 18) show in general that our approach (C-MCF) has better performances compared to the other methods, except in the link utilization where HPAC has less values. This is because HPAC uses Gomory–Hu transformation to optimize and reduce the network topology. Besides to this, C-MCF outperforms S-MCF thanks to the caching functionality, the caching nodes are set to be near to the consumers. Thus, they respond to all incoming requests preventing them from reaching the data-source node, this leads to shorten the distance in the transmission. In experiments 1 and 3 (Figs. 15a and 17a), there is a decrease in the link utilization, because there are 05 clients selected, and the scale of the network is increased each time. This percentage is subject to the total number of nodes in each step. Besides to this, C-MCF and S-MCF construct different SPTs (Shortest-Paths Tree) that uses only the best paths. For

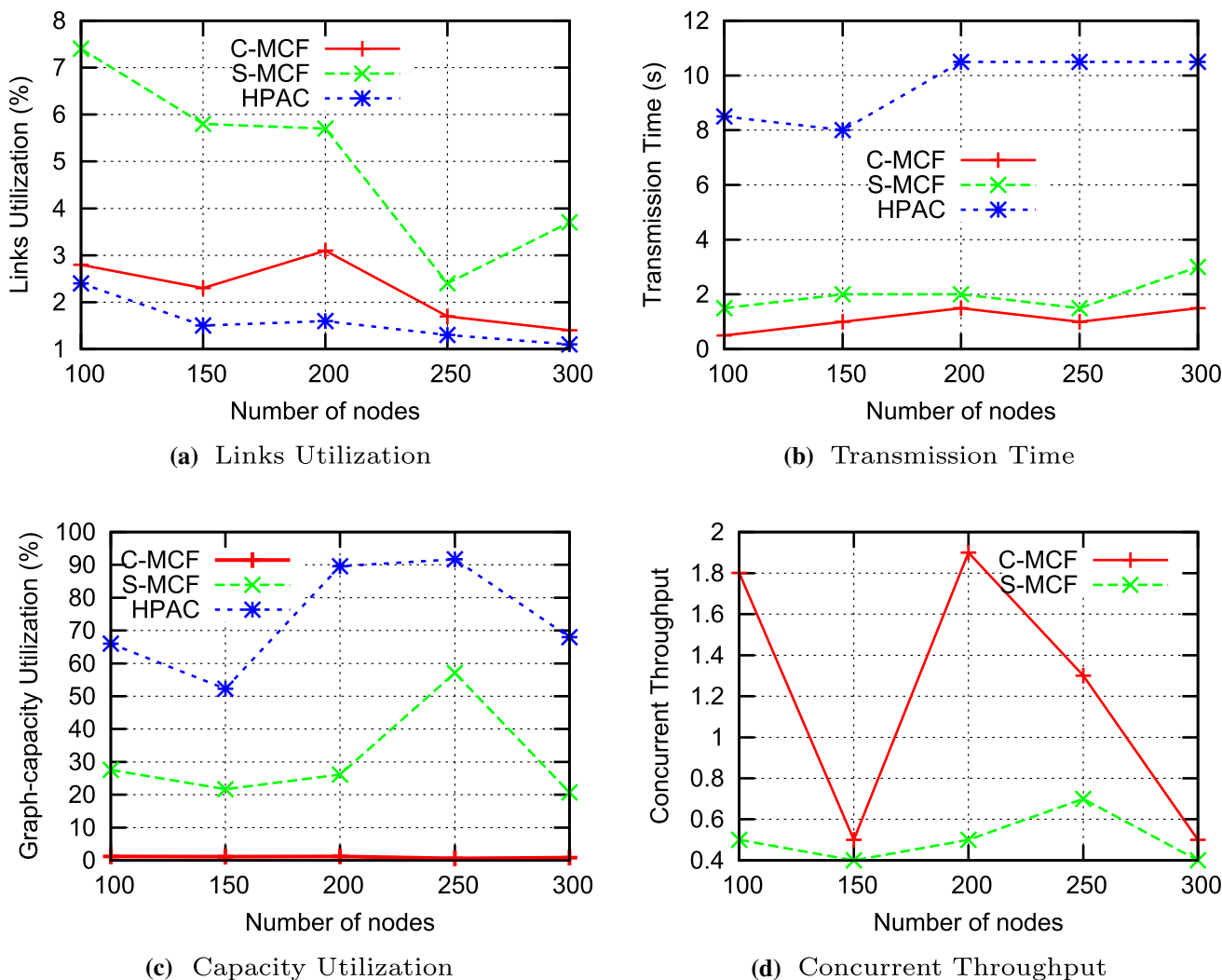


Fig. 17 Evaluation of the approaches (Experiment 3)

experiments 2 and 4 (Figs. 16 and 18), we maintained one topology and increased the number of consumers. Obviously, this leads to add new shortest paths to the SPT in order to link these destination nodes with the data source.

When it comes to the time cost (Figs. 15b, 16b, 17b, 18b), C-MCF outperforms S-MCF and HPAC due to the use of caching feature. Again, the caching placement is selected to be close to the clients (as shown in Fig. 14b), this means less hops and less transmission time. The concurrency of multi-commodity is the second advantage. Instead of sending each commodity in separate data packets, C-MCF and S-MCF exploit the network capacity (i.e., bandwidth) by encapsulating different commodities within common packets. Hence, the needed packets for transmission are reduced. This results in a shorter time.

The three methods are also evaluated in terms of capacity utilization (Figs. 15c, 16c, 17c, 18c). Again, we distinguish two scenarios: the variation of utilization when the graph

scales up, and when there is an increasing number of consumers. We can clearly see that C-MCF has minimal values compared to S-MCF and HPAC, this is due to the construction of small MCF SPTs rooted at each caching node. The selection of the caching nodes has contributed to reduce the consumption of the overall network capacity.

The final criterion in this assessment is the concurrent throughput (Figs. 15d, 16d, 17d, 18d). In this part, we examine only the multicommodity methods (i.e., C-MCF and S-MCF) in different scenarios: variation in the network scale, variation in the number of clients, and their Requests. The results show that C-MCF has a higher throughput than S-MCF. We also remark some fluctuations, because the caching placement is set according to many factors: the scale of the network, the capacity of the shortest paths. Also, the size of the data requested by the clients impact the concurrent throughput. The capacity of MCF SPT plays a major role in this regard since the MCFP constraints must be verified.

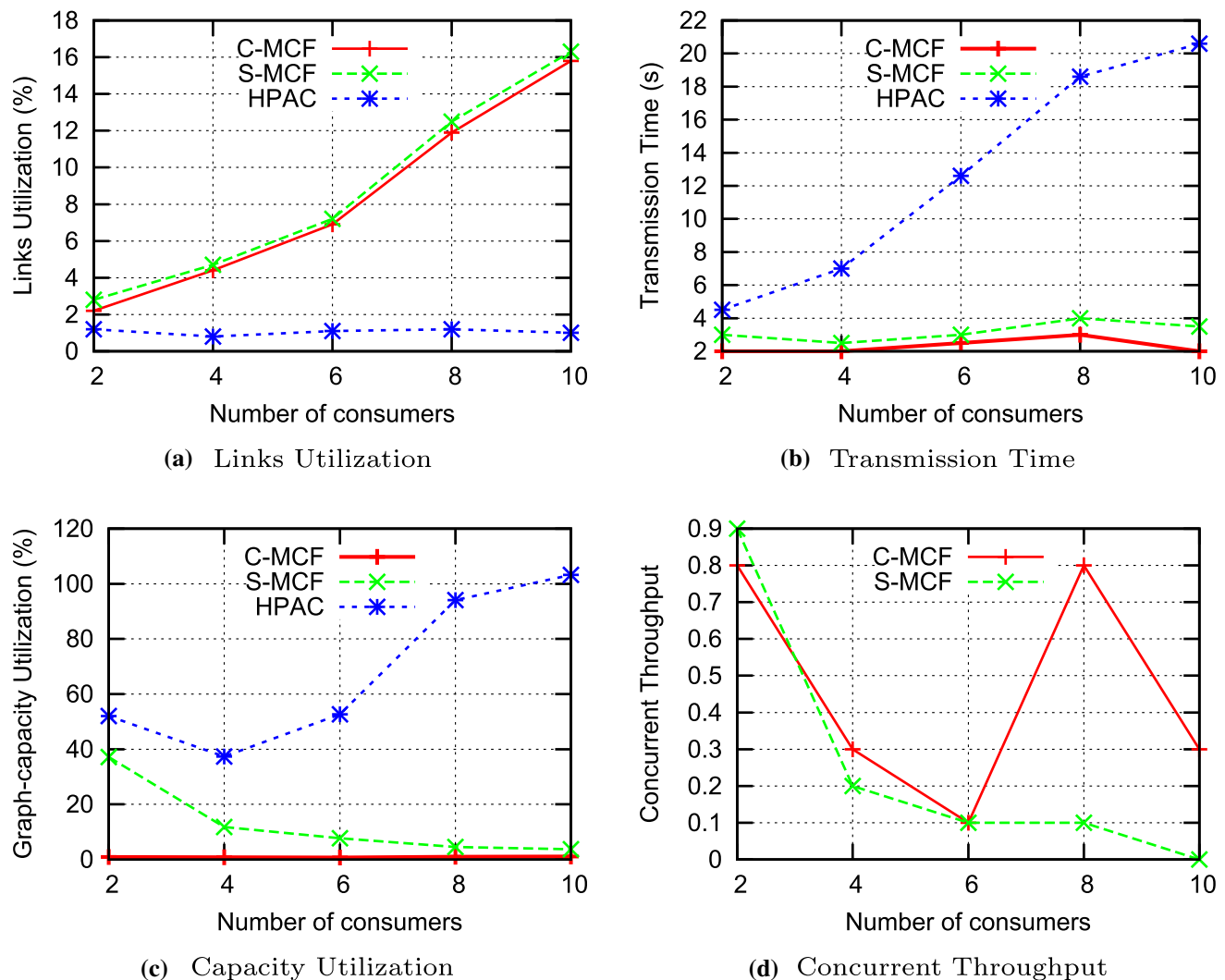


Fig. 18 Evaluation of the approaches (Experiment 4)

8 Conclusion and Perspectives

In this paper, we overviewed the routing in IP networks and its constraints to support the next-generation of data. We chose the multicommodity optimization field, especially the MCFP approach and we presented some related works. In order to cope with the massive multimedia data traffic, we developed two promising solutions to improve the quality of data routing in Multimedia IoT networks. We firstly proposed a multicommodity protocol based on the MCFP paradigm, followed by a second contribution that improves the routing much better by integrating the content-caching feature in the MCF routing. This feature is based on a novel placement strategy. Both solutions have been extensively evaluated and compared showing good results and improvement over the state of the art.

As future works, there are many interesting ideas as a follow-up to this work. The machine learning (ML) has

shown a great impact in terms of performance optimization in various domains over the past years [54]. Particularly, the reinforcement learning (RL) that has many applications. One of the main advantage of the RL is the efficient adaptability to any change in the environment. This technique has been applied in our previous work [55] to find quickly the shortest paths in dynamic IoT networks. In general, the ML is specifically used for three tasks: (i) Prediction, (ii) Classification, and (iii) Decision making. For our upcoming works, the first technique will be used to predict which router will be down in the future, in order to reroute the concurrent flows beforehand to the best shortest-paths. The Classification technique can be used to classify the routed data according to its rate of circulation within the network; also in order to alleviate the bandwidth consumption and improving the rapidity of routing, the popular data (issued from the classification process) will be migrated and stored in specific routers. The selection of these routers is performed using the third ML technique

that is called “Decision making” after executing the learning phase. Moreover, and as depicted early in Fig. 1 (the right intersection), we will tackle the routing improvement in the ICN network using MCFP because it is the next generation of the Internet that will handle huge content traffics at a very large scale with billion of devices. Also, we intend to investigate the integration of SDN technology with our solutions. Lastly but not least, many evaluations and comparisons are planned with other approaches such as segment routing in the context of ICN networks.

Acknowledgements This research was supported financially by a P.N.E (Programme National Exceptionnel) scholarship given by the Algerian Ministry of higher education (MESRS) and scientifically by the research project PRFU with code = C00L07UN310120220008.

References

- Sobin, C.C.: A Survey on Architecture Protocols and Challenges in IoT. Springer, US (2020)
- Bayılmış, C.; Ali Ebleme, M.; Çavuşoğlu, Ü.; Küçük, K.; Sevin, A.: A survey on communication protocols and performance evaluations for internet of things. *Digital Commun. Netw.* (2022)
- Shambour, M.K.; Gutub, A.: Progress of iot research technologies and applications serving hajj and umrah. *Arab. J. Sci. Eng.* **47**(2), 1253–1273 (2022)
- Conti, M.; Gangwal, A.; Hassan, M.; Lal, C.; Losiouk, E.: The road ahead for networking: a survey on icn-ip coexistence solutions. *IEEE Commun. Surv. Tutor.* **22**(3), 2104–2129 (2020)
- Zhang, J.; Wang, X.; He, Q.; Huang, M.: Teap: Traffic engineering and alr policy based power-aware solutions for green routing and planning problems in backbone networks. *Comput. Commun.* **173**, 27–44 (2021)
- Shahrokhi, F.; Matula, D.W.: The maximum concurrent flow problem. *J. ACM* **37**(2), 318–334 (1990)
- Arshad, S.; Azam, M.A.; Rehmani, M.H.; Loo, J.: Recent advances in information-centric networking-based internet of things (icn-iot). *IEEE Internet Things J.* **6**(2), 2128–2158 (2019)
- Jazebi, S.J.; Ghaffari, A.: Risa: routing scheme for internet of things using shuffled frog leaping optimization algorithm. *J. Ambient. Intell. Humaniz. Comput.* **11**, 4273–4283 (2020)
- Seyfollahi, A.; Ghaffari, A.: A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks. *Comput. Netw.* **179**, 107368 (2020)
- Azari, L.; Ghaffari, A.: Proposing a novel method based on network-coding for optimizing error recovery in wireless sensor networks. *Indian J. Sci. Technol.* **8**, 859–867 (2015)
- Lim, W.S.; Jung, H.; Chun, W.: Design of scalable link-state routing in future internet. In: 2015 International Conference on Information and Communication Technology Convergence (ICTC), pp. 589–594. (2015)
- Ospf version 2: Rfc 2328. <https://tools.ietf.org/html/rfc2328>, [Accessed 2020 July 7]
- Ospf for ipv6: Rfc 5340. <https://tools.ietf.org/html/rfc5340>, [Accessed 2020 July 7]
- Cisco annual internet report (2018-2023) white paper. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>, [Accessed 2020 Jun 1]
- Garg, N.; Könemann, J.: Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In: Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280), pp. 300–309. (1998)
- Fleischer, L.K.: Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discret. Math.* **13**(4), 505–520 (2000)
- Bienstock, D.; Raskina, O.: Asymptotic analysis of the flow deviation method for the maximum concurrent flow problem. *Math. Program.* **91**(3), 479–492 (2002)
- Chen, Y.; Zhang, Y.; Yuan, M.: An improved flow deviation method for the maximum concurrent flow problem. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1, pp. 1179–1182. (2020)
- Zhang, Y.; An, X.; Yuan, M.; Bu, X.; An, J.: Concurrent multipath routing optimization in named data networks. *IEEE Internet Things J.* **7**(2), 1451–1463 (2020)
- Salimifard, K.; Bigharaz, S.: The multicommodity network flow problem: state of the art classification, applications, and solution methods. *Oper. Res.* 1(0123456789). (2020)
- Orangi-Fard, N.: Maximum concurrent flow problems and p-modulus. PhD thesis, Kansas State University, (2020)
- Bauguion, P.O.; Ben-Ameur, W.; Gourdin, E.: Efficient algorithms for the maximum concurrent flow problem. *Networks* **65**(1), 56–67 (2015)
- Aboodi, A.; Wan, T.; Sodhy, G.: Survey on the incorporation of ndn/ccn in iot. *IEEE Access* **7**, 71827–71858 (2019)
- Multicast extensions to ospf: Rfc 1584. <https://tools.ietf.org/html/rfc1584>, [Accessed 2020 July 7]
- D’Arienzo, M.; Romano, S.P.: Gospf: An energy efficient implementation of the ospf routing protocol. *J. Netw. Comput. Appl.* **75**, 110–127 (2016)
- Wang, L.; Hoque, A.K.M.M.; Yi, C.; Alyyan, A.; Zhang, B.: Ospf: An ospf based routing protocol for named data networking ndn. *Tech. Rep. UCLA.* (2012)
- Wang, I.-lin: Multicommodity network flows?: a survey, part I?: applications and formulations. *Int. J. Oper. Res.* **15**(4), 145–153 (2018)
- Barnhart, C.; Krishnan, N.; Vance, P.H.: Multicommodity flow problems. pp. 2354–2362. Springer US, Boston, MA (2009)
- Madry, A.: Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. In: Proceedings of the Forty-second ACM Symposium on Theory of Computing, STOC ’10, pp. 121–130. New York, NY, USA (2010). ACM
- Karakostas, George: Faster approximation schemes for fractional multicommodity flow problems. *ACM Trans. Algorithms* **4**(1), 13:1-131:7 (2008)
- Young, N.E.: Randomized rounding without solving the linear program. In: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’95, pp. 170–178, Philadelphia, PA, USA, (1995). Society for Industrial and Applied Mathematics
- Radzik, T.: Fast deterministic approximation for the multicommodity flow problem. *Math. Program.* **78**(1), 43–58 (1996)
- Klein, P.; Plotkin, S.; Stein, C.; Tardos, E.: Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM J. Comput.* **23**(3), 466–487 (1994)
- Goldberg, A.V.: A natural randomization strategy for multicommodity flow and related algorithms. *Inf. Process. Lett.* **42**(5), 249–256 (1992)
- Nauman, A.; Qadri, Y.A.; Amjad, M.; Zikria, Y.B.; Afzal, M.K.; Kim, S.W.: Multimedia internet of things: a comprehensive survey. *IEEE Access* **8**, 8202–8250 (2020)
- Zikria, Y.B.; Afzal, M.K.; Kim, S.W.: Internet of multimedia things (iomt): Opportunities, challenges and solutions. *Sensors*, 20(8). (2020)



37. Affinito, A.; Botta, A.; Ventre, G.: The impact of covid on network utilization: an analysis on domain popularity. In: 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6. (2020)
38. De Cicco, L.; Manfredi, G.; Palmisano, V.; Mascolo, S.: A multi-commodity flow problem for fair resource allocation in multi-path video delivery networks. *IFAC-PapersOnLine*, 53(2):7386–7391, (2020). 21th IFAC World Congress
39. Djaker, A.; Kechar, B.; Afifi, H.; Moun gla, H.: A maximum concurrent flow technique for optimized data routing in iot architectures. In: 2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC), pp. 1–2. (2020)
40. Jupyter project. <https://jupyter.org>, [Accessed 2020 July 15]
41. Networkx library for complex networks. <https://networkx.github.io/>, [Accessed 2020 July 15]
42. Designing scalable ospf design. <https://www.ciscopress.com/articles/article.asp?p=1763921&seqNum=6>, [Accessed 2020 July 7]
43. Di Caro, G.; Dorigo, M.: Antnet: Distributed stigmergetic control for communications networks. *J. Artif. Int. Res.* **9**(1), 317–365 (1998)
44. Doyle, J.; Carroll, J.D.H.: Routing TCP/IP. In: *CCIE Preparation Guides Series*, vol. 1. Macmillan. (1998)
45. Ibn Khedher, H.; Hadji, M.; Abousabea, E. M. A.; Afifi, H.; Kamal, A. E.: Scalable and cost efficient algorithms for virtual CDN migration. In: *LCN 2016 : 41st IEEE Conference on Local Computer Networks*, pp. 112 – 120. Dubai, United Arab Emirates, November (2016). IEEE Computer Society
46. Jeon, S.; Jung, K.; Chang, H.: Fully distributed algorithms for minimum delay routing under heavy traffic. *IEEE Trans. Mob. Comput.* **13**(5), 1048–1060 (2014)
47. Chandra, S.; Shrivastava, U.; Vaish, R.; Dixit, S.; Rana, M.: Improved-antnet: Aco routing algorithm in practice. In: 2009 11th International Conference on Computer Modelling and Simulation, pp. 25–29. (2009)
48. Miller, G.; Naor, J.: Flow in planar graphs with multiple sources and sinks. *SIAM J. Comput.* **24**(5), 1002–1017 (1995)
49. Lissner, A.; Mahey, P.: *Multicommodity Flow Problems and Decomposition in Telecommunications Networks*, pp. 241–267. Springer US, Boston, MA (2006)
50. Yamamoto, Y.; Takemasa, J.; Koizumi, Y.; Hasegawa, T.: A packet-level caching algorithm for mitigating negative effects caused by large objects in icn networks. *IEEE Access* **8**, 108247–108259 (2020)
51. Hua, Y.; Guan, L.; Kyriakopoulos, K.G.: A fog caching scheme enabled by icn for iot environments. *Futur. Gener. Comput. Syst.* **111**, 82–95 (2020)
52. Che, X.; Ip, B.; Lin, L.: A survey of current youtube video characteristics. *IEEE Multimedia* **22**(2), 56–63 (2015)
53. Cheng, X.; Liu, J.; Dale, C.: Understanding the characteristics of internet short video sharing: a youtube-based measurement study. *IEEE Trans. Multimedia* **15**(5), 1184–1194 (2013)
54. Begam, S.; Sangeetha, M.; Shanker, N.: Load balancing in dcn servers through sdn machine learning algorithm. *Arabian J. Sci. Eng.* **47**, 1423–1434 (2021)
55. Djaker, A.; Kechar, B.; Ibn-Khedher, H.; Moun gla, H.; Afifi, H.: Scalable and cost efficient maximum concurrent flow over iot using reinforcement learning. In: 2020 International Wireless Communications and Mobile Computing (IWCMC), pp. 539–544. (2020)

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.