IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 15, 2021, accepted December 10, 2021, date of publication January 3, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3139936

Multipath Routing Over Star Overlays for Quality of Service Enhancement in Hybrid Content Distribution Peer-to-Peer Networks

MEHMET KARAATA, ANWAR AL-MUTAIRI, AND SHOUQ ALSUBAIHI
Department of Computer Engineering, Kuwait University, Safat 13060, Kuwait
Corresponding author: Mehmet Karaata (mehmet.karaata@ku.edu.kw)

ABSTRACT *Content Delivery Networks* (CDN's) have emerged as a flexible and decentralized solution to maintain and transfer large volumes of data. CDN's are distributed systems that maintain a distributed storage on a large number of servers at various locations distributed all over the world and a service network system for dissemination of content such as videos and software with high content dissemination efficiency, enhanced QoS metrics, and reduced network load. In the wake of enormous growth in live video streaming traffic on the Internet, CDN's face challenges in meeting video traffic demands of users. As a remedy, *hybrid CDN-P2P networks* are being deployed to allow P2P networks to share the content delivery load of CDN's providing the reliability and the performance of the CDN's, and the scalability and the low cost of P2P networks. In this paper, by simulation under a realistic model, we show that multipath routing in star overlay networks achieves a high degree of load balancing, scalability, throughput enhancement, and reduces buffer requirements and network bottlenecks. As these algorithmic properties are highly desirable for hybrid CDN-P2P networks, we establish the viability of the star overlay networks as an edge network for hybrid CDN-P2P networks to meet their content delivery quality of service requirements.

INDEX TERMS Edge networks, hybrid CDN-P2P networks, multipath routing, overlays, star networks.

I. INTRODUCTION

Content Delivery Networks (CDN's) have emerged as a flexible and decentralized solution to transfer large volumes of data primarily for video-on-demand, personal live streaming, software download and DDOS protection [1], [2]. CDN's are distributed systems that maintain a distributed storage on a large number of servers at various locations distributed all over the world and a service network system for dissemination of content such as videos and software with high content dissemination efficiency, enhanced QoS metrics for end-users, and reduced network load. CDN's have been proposed by the Internet Engineering Task Force (IETF) [3] as a content network to cope up with the enormously growing demand for video and content distribution. CDN's benefit not only the end users, but also the content providers and the Internet service providers (ISP's) who deploy CDN's servers in their networks [4]. With CDN's, the end users experience higher QoS as the download latency and the *bandwidth* are improved, where the bandwidth refers to the maximum rate or amount of data transfer between two endpoints in a given amount of time. In addition, with CDN's, the content providers can offer larger volumes of reliable services, and the ISP's enjoy reduced traffic on their backbone servers. Professionally managed and geographically distributed infrastructure of CDN's is highly reliable, available and provides high quality service. However, CDN's require considerable investments for deployment, scaling up, and management of geographically distributed servers [5].

In the wake of enormous growth in live video streaming traffic on the Internet, CDN's face challenges in meeting video traffic demands of users. As a remedy, *hybrid CDN-P2P networks* are being deployed to allow P2P networks to share the content delivery load of CDN's providing the reliability and the performance of the CDN's, and the scalability and the low cost of P2P networks [6]. In such a network, each peer may select one of the closest CDN edge servers to receive content available in the CDN and this edge server is considered as a peer in the P2P network. In a hybrid CDN-P2P network, whenever there is sufficient network and storage capacity in the P2P network component, peers distribute shares of content among themselves using

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

techniques such as centrally managed swarming [7]. Upon a content request by a user, if there are peers near the user with free upload capacity to deliver the content while maintaining the expected quality, user is served by the peers; and users are served directly from the CDN servers, otherwise.

Huang and Zhang [8] present a feasibility study of a novel peer-to-peer architecture for live video streaming. The proposed architecture manages a P2P overlay to deliver audio/video streams through the use of online social networks to retrieve user information and relationships between them in order to improve overlay and stream management. However their proposal does not use hybrid CDN-P2P, a specific overlay topology and multipaths. Commercial hybrid peer-to-peer video delivery systems such as *CDN Mesh Delivery and Peer5* exist providing media delivery with improved performance, increased reliability and expanded reach for broadcasters while delivering more reliable and more scalable for end users by intelligently multi-sourcing video delivery from both the CDN and a P2P network of end users [9]–[11].

Hybrid CDN-P2P networks have recently emerged as an economically viable alternative to traditional content delivery networks. The feasibility studies conducted by several large content providers suggested a remarkable potential for hybrid CDN-P2P networks to reduce the burden of user requests on content delivery servers [12]. Subsequently, several commercial hybrid CDN-P2P networks deployments have been introduced [12]. However, there are numerous commercial and technical challenges that negatively affect the prospects of industrial hybrid CDN-P2P solutions. In order to enhance the content distribution services, approaches such as hybrid CDN-P2P networks have been designed and studied to allow content distribution to scale or adapt to the bandwidth of data transfer. A hybrid CDN-P2P network requires all potential parallel paths in its P2P component to be discovered and utilized upon demand and the load related parameters. Additional challenges include the reliability, availability and scalability related issues of peer-to-peer edge networks, the lack of incentive mechanisms for peer participation, and copyright issues. The reliability issues related to the P2P edge network stem from insufficient bandwidth, lack of required degree of *network throughput*, *load balancing*, buffering issues, and the presence of *network bottlenecks*. Network throughput refers to the amount of data transferred in the network during an interval while load balancing refers to the even distribution of messages among the peers in the routing process. Network bottlenecks refer to the limitations of some network resources such as buffers at peers and channel capacities that limit the network capacity to transfer content in a timely manner.

In addition, a hybrid CDN-P2P network cannot cope up with flash crowd content and heavy content demand. Research has shown that viewers are not patient enough to wait if the start-up delay is longer than a few seconds [13]. Measurements given in [14] also confirm that users very often suffer from video re-buffering or more than five seconds start-up delay. As a result, users tend to drop videos if they frequently stop, freeze, or experience quality changes during the service period [15].

The massive volume of content traffic due to the growth in mobile Internet, computer networks, ultra-high definition videos, and user generated content presents unsurpassed challenges to CDN's. To cope up with this enormous content demand, network service and content providers take advantage of CDN's as they are widely regarded as a viable approach to successfully, and efficiently manage content traffic. Nevertheless, efficiency and other metrics of quality of major available methods for content routing are insufficient to meet the current demand. For that purpose, sophisticated content access and dissemination approaches, particularly multimedia streaming, utilize *multipaths* to provide the content with expected quality by increasing network bandwidth, reducing network congestion and latency.

It is known that most of the challenges related to service quality can be met through the appropriate selection of an *overlay structure* providing sufficient number of multipaths between communication endpoints. A *peer-to-peer overlay network* is a virtual or logical network of overlay peers connected by virtual or logical links and constructed on the top of a physical network called *underlay*. An ideal overlay network with an appropriate number of multipaths between communication endpoints increases the network bandwidth while evenly balancing the network load among peers and links of the network, reduces network bottlenecks, increases system throughput, and provides fair service to users. For instance, star overlay networks [16] and their variations [17] provide a large number of parallel paths, a small graph diameter, scalable lookup service for the peers participating in Peer-to-Peer (P2P) networks and a small degree compared to conventional hypercubic DHTs such as Chord and Kademlia. Existing implementations of hybrid CDN-P2P networks have the following shortcomings that limit their reliability, availability and quality of service. First, default best-effort Internet routing results in the absence of end-to-end QoS. Second, existing routing algorithms primarily focus on router and link factors on a single path and thus do not effectively utilize the available network through the use of multipaths. Third, routing policies primarily focus on local knowledge in an individual autonomous system, lacking a network-wide view of topology or traffic to optimize routing with respect to load balancing, throughput, bandwidth, and delay requirements. Fourth, existing hybrid CDN-P2P overlay topologies provide no multipaths or only a limited number of multiple disjoint paths between endpoints that can be readily utilized for bandwidth enhancement or load balancing in P2P networks. Fifth, high-definition video streaming and other forms of content delivery do not scale well to support a large number of end-users, but achieving scalability is very hard since the communication cost and the load of some servers may be extremely high when the number of users is large [18]. In addition, growing demand on media steaming and other content distribution applications have led to more stringent quality of service requirements including high bandwidth, highly reliable and scalable service many of which depend on the load balancing and multipath routing ability of the routing algorithms [19]. Hybrid CDN-P2P networks have been claimed to meet some of these challenges where the P2P component can facilitate the scalability, bandwidth enlargement and the low cost, distributes the system's load to all participants, handles flash crowd and reduces the load on CDN servers while the CDN component ensures the reliable and high-quality service. However, usage of multipath routing algorithms and overlay networks with a large number of multipaths such as star networks for hybrid CDN-P2P networks for real world applications to address the above shortcomings have been neither proposed nor evaluated.

As a remedy, in this paper, by simulation under a realistic model, we show that multipath routing in star overlay networks achieves a high degree of load balancing, scalability, throughput enhancement, and reduces buffer requirements and network bottlenecks. As these algorithmic properties are highly desirable for hybrid CDN-P2P networks, we establish the viability of the star overlay networks as an edge network for hybrid CDN-P2P networks to meet their content delivery quality of service requirements. In particular, we simulated the multipath routing algorithm of Karaata and Alsulaiman [16] under a realistic model including the essential aspects that are not considered in [16] for a practical implementation of the algorithm such as buffer requirements of peers, limiting channel capacities, concurrent transmission of multiple content from multiple sources, pipelining/ interleaving of multiple messages over the same set of multipaths between two endpoints and message drops. Through our simulation, we established the following. First, our simulation results demonstrate that the star overlay with multipath routing balances network load irrespective of the network size and demand. Second, we show that as the content delivery demand increases, network throughput linearly increases. This demonstrates that the star overlay networks have sufficiently many multipaths between all pairs of endpoints whose utilization allows network throughput to increase significantly. Third, the experiments show that the star overlay networks do not require larger buffer sizes as the throughput increases for small and large size networks. This demonstrates the high degree of scalability of the overlay network with the multipath routing for hybrid CDN-P2P networks. The same also show that the overlay with the multipath routing does not lead to network bottlenecks. Fourth, our simulation results show that the star overlay networks with the multipath routing algorithm of [16] delivers content from a source to a destination peer over multiple overlay paths in at most $D(S_n) + 4$ cycles/rounds.

The rest of the paper is organized as follows. P2P overlay networks and multipath routing are presented in Section II providing the required background and terminology. Section III presents a brief overview of the inherently-stabilizing routing algorithm for star P2P overlay networks [16] that is simulated and evaluated in this paper. The network simulation model is described in Section IV. Section V presents the simulation results related to the message propagation delay, network throughput, buffer requirements, load balancing and scalability of the inherently self-stabilizing multipath routing protocol for hybrid CDN-P2P networks. Section VI concludes the paper and features some future research directions.

II. PRELIMINARIES

A. P2P OVERLAY NETWORKS

CDN architectures often rely on virtual overlay networks constructed on the generic IP protocol to solve performance problems related to network congestion and to improve web content accessibility in a cost-effective manner [4], [20].

The primary purpose of a P2P component in a hybrid CDN-P2P network is collaboration among peers to facilitate sharing resources and services to enhance the combined network. The quality of sharing of services and resources heavily relies on the available network and the routing protocols that facilitate peer-to-peer communication. Routing protocols often enhance a peer-to-peer network via increasing the network bandwidth, eliminating network bottlenecks through load balancing, and reducing message propagation delays.

Peer-to-peer (P2P) overlay networks were initially devised for file sharing; however, later, they have become popular for content sharing, media streaming, telephony applications such as the P2PTV and PDTP protocols. Numerous other widely used P2P applications also exist. For instance, some proprietary multimedia applications use a peer-to-peer network along with streaming servers to stream audio and video to their clients. Bitcoin and alternatives such as Ether, Nxt and Peercoin are all peer-to-peer-based digital cryptocurrencies [21]–[23]. Dalesa is a peer-to-peer web cache for LAN based on IP multicasting [24]. P2P-based search engines such as FAROO also exists [25]. Filecoin is a P2P-based open source, public cryptocurrency and digital payment system intended to be a blockchain-based cooperative digital storage and data retrieval method [26]. I2P is another P2P-based application built over an overlay network to browse the Internet anonymously [27].

B. MULTIPATH ROUTING

There are two types of routing protocols used for the collaboration among peers, namely *single path routing* and *multiple path routing*. In a single path routing protocol, throughout the session for sharing resources between peers, a single path is used between the sender and the receiver peers. When a single path is used by the routing algorithms, other potential paths between the communicating peers are neither constructed nor utilized to enhance communication. This does not allow single path routing to significantly widen network bandwidth, avoid network bottlenecks, balance the network load and reduce propagation delays. Whereas, in a multipath routing, the same message is split into multiple shares and sent simultaneously over multiple paths established between a pair of peers. Usage of multipath routing clearly enhances the communication bandwidth between the peers by using bandwidth facilitated by the available multipaths, reduces the message propagation delays of large size messages as message shares sent simultaneously over multiple paths requires less propagation delay compared to those sent in sequence over a single path, becomes more tolerant to network failures than traditional single path approaches and improves the security of message transmission, balances network load and reduces network bottlenecks caused by heavy usage of limited network bandwidth provisioned by a single path routing. Load balancing is a very desirable feature since it promotes *availability*, *scalability* and reduces the occurrence of bottlenecks in the overlay. Availability means that the network is available as it is operating correctly at any given time while scalability means being able to handle the growth in size and the increase in future load.

Multipath routing is already used in various networks. For example, Named Data Networks (NDN's) inherently provide a flexible forwarding plane for multi-source and multipath communications [28]. In NDN's, hosts utilize multipaths to obtain data from multiple content providers via multiple paths, which is different from IP multipath routing [29]. In VANET's, multihop and multipath routing exploiting several paths is proposed to achieve faster content retrieval [30]. Content delivery networks also utilize multipaths in multipath pre-caching mechanisms in which the edge server would parse the requested content and then distribute requests to other edge servers to download content from the origin server simultaneously for accelerating the download speed [31]. In [32], authors propose a video delivery system involving CDN's that use bandwidth aggregation of multiple ISP's simultaneously via multipath content delivery. The paper suggests that the multipath approach increases the average quality of service at the expense of ISP's that experience disproportional congestion increases under heavy load because multipath approach is able to scrounge the last bits of available bandwidth on every ISP reducing the number of served requests.

III. INHERENTLY-STABILIZING MULTIPATH ROUTING ALGORITHM FOR STAR P2P OVERLAY NETWORKS

In this section, we present a brief overview of the inherently-stabilizing routing algorithm for star P2P overlay networks [16] that is simulated and evaluated in this paper. The algorithm proposed by Karaata *et al.* is for routing messages over all disjoints paths between two peers in a star P2P overlay network. In an n-dimensional star network, the algorithm is capable of routing up to $n-1$ message shares simultaneously. The algorithm is optimal in terms of the length of the disjoint paths. Due to being inherently-stabilizing, the algorithm can autonomously start in any state and can always recover from *transient faults*. A transient fault refers to a fault that perturbs the state of a process but not its program. In addition, as the algorithm is inherently self-stabilizing, faults perturbing variables of the system are masked and thus the execution of the algorithm is not affected by arbitrary initialization and transient faults.

The simulation model of the inherently-stabilizing routing algorithm is built for an undirected n-dimensional star graph network $S_n = (V, E)$ where V is the set of $n!$ vertices each of which corresponds to a peer in the peer to peer network such that each permutation of symbols $1, 2, 3, \ldots, n$ makes up an id of a distinct vertex while E is the set of symmetric edges. Each vertex has $n-1$ neighbors connected through distinct edges. Two nodes are connected by an edge iff the id of one can be obtained from the other by interchanging its first symbol with any other symbol. For example, the i^{th} neighbor v of s refers to the neighbor of peer s whose *id* is obtained by swapping symbols at Position 1 and i of s. Thus, the number of edges in S_n is given by $L = {}^{(n-1)n!}/_{2}$ [33].

A. THE ROUTING PROTOCOL MODEL AND INTERFACE

Since an n-dimensional star graph is used where there exist $n-1$ disjoint paths between any pair of vertices, a message can be transferred between a source peer and a destination peer using $n-1$ disjoints paths; hence, each message M to be transferred is split into $n-1$ message shares, i.e., $M = m_0, m_1, m_2, \ldots, m_{n-2}$. A protocol called the *application protocol* is assumed to exist at each peer that sends messages from a source peer to a destination peer using node-disjoint paths algorithm over all node-disjoint paths.

To implement the interface between the application protocol and the node-disjoint algorithm at each peer, the algorithm maintains two implicit buffers for each peer, namely, the *implicit input buffer* and the *implicit output buffer*. When the application protocol at peer s wants to send message M to destination peer d, it places both the message and destination id d in the implicit input buffer of the peer. Subsequently, upon discovering message M in its input buffer, the routing algorithm at peer s receives message M by removing the input from the input buffer of s. The routing algorithm later uses action **output(m)** to place each message share m in the output buffer of d to make it available to the application protocol at destination d. It is assumed that between the execution of two output actions, the application protocol removes the content of the output buffer. As each peer contains both the input and output buffers, the algorithm allows each peer to act as a source peer or a destination peer. At any point in time, the input buffer contains at most a single sequence of $n-1$ message shares and a single destination *id* while the output buffer contains at most a single message share.

Each peer also contains an implicit routing buffer that is used in routing the input message share by the peers on the path from the source peer to the destination peer. This buffer holds at most a single share of each input message with destination *id*, and the distinguishing position *lsp* (last swap position) that holds the first symbol of destination process to ensure node-disjointness. The algorithm assumes asynchronous message passing model where a message share moves between neighboring peer buffers after an arbitrary but finite propagation delay. The transmission

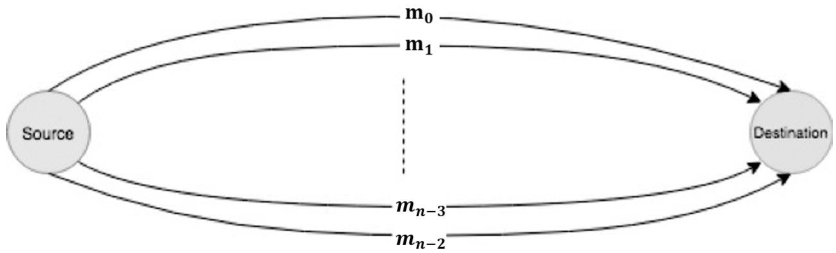

FIGURE 1. Multiple path routing.

of an input message M is always completed in at most $D(S_n)+4$ rounds/cycles, where $D(S_n)$ is the distance between the source and destination where $D(S_n) = \lfloor ^{3(n-1)}/_2 \rfloor$. Therefore, the algorithm has a time complexity of $D(S_n)+4$ rounds which is also the length of the longest path traversed by a message.

B. ROUTING ALGORITHM

When a source peer s receives message M and destination peer id d, s divides the message into $n-1$ shares and maps each of its neighbors to a distinct neighbor of the destination peer d and then sends each share to one of its mapped neighbors. The message shares are then routed between pairwise mapped neighboring peers of s and d over node-disjoint paths. When a message share m is received by a neighbor of peer d, it is sent to destination d. To ensure that all the paths between pairwise mapped neighbors of s and d are disjoint, the algorithm employs the method given next.

In the routing process, to rout message share m from peer v to a neighboring peer, the first symbol $v[1]$ is swapped with another symbol $v[j]$, where $2 < j \leq n$, to determine the id of the neighboring peer to send m. Recall that the id of peer $v \in V_n$ is a permutation over *1,2,3,...,n* where $v[i]$ denotes the i^{th} symbol of v and $1 \leq i \leq n$. The schemes for determining the value of the swap position j by the source peer s and the other peers differ. Source peer s first splits the input message into $n-1$ message shares to sent to $n-1$ neighbors. Subsequently, for each message share m_i, s swaps $s[1]$ with distinct symbol $s[i]$ to determine the neighbor to send message share m_i, where $1 < i \leq n$. For each message share m_i, peer s also determines a distinct position *lsp* (last swap position) to send along with m_i to the i^{th} neighbor, where $1 < lsp \leq n$. Once the i^{th} neighbor of s receives the message share m_i, it places $d[1]$ in position *lsp*, if not already there, and maintain it there until the last swap. This serves two purposes. First, as $d[1]$ is placed and kept in distinct position *lsp* for each path, process id's on each path are distinct from those on other paths leading to the construction of $n-1$ node-disjoint paths. Second, for the same reason, neighbor w of d can be reached such that d is obtained by swapping $w[1]$ and $w[lsp]$. Therefore, *lsp* of m_i determines the neighbor w of d that will receive m_i. In order to place $d[1]$ in position *lsp*, peer v that receives message share first places $d[1]$ in position $v[1]$ by swapping $v[1]$ with the position in v that holds the value of $d[1]$. Then, $d[1]$, which is now stored in $v[1]$, is swapped with symbol $v[lsp]$. Once $d[1]$ is placed in position *lsp* on all paths, each peer v on the constructed paths determines the id of the next peer by swapping symbols in position $v[1]$ and $v[k]$ where k denotes the position of $v[1]$ in d, that is, $v[1] = d[k]$. Note that this swapping is only done when $v[1] \neq d[lsp]$, otherwise, $v[1]$ is swapped with an *unsorted* position instead of *lsp* to keep $d[1]$ in position *lsp*. The i^{th} symbol of v is said to be sorted if $v[i] = d[i]$; unsorted, otherwise. This swap is repeated until reaching a neighbor w of d which completes the routing peer by swapping position $w[1]$ with $w[lsp]$ to reach d.

The proposed inherently-stabilizing routing algorithm in [16] merely provides a distributed algorithm for routing a single message over multipaths in star overlay networks with desirable features such as inherent stabilization and stabilization. In [16], an abstract model is assumed where essential details for a practical implementation of the algorithm under a realistic model such as buffer requirements of peers limiting channel capacities, concurrent transmission of multiple messages from multiple sources, pipelining/interleaving of multiple messages over the same set of multipaths, and *message drops* are not considered. In addition, each peer is assumed to have a single input and a single routing buffer which are relaxed in our paper. A message drop refers to an event in which the message share arrives at a peer whose buffer is full. In addition, the experimental work to show that the algorithm is correct and it improves throughput, increases bandwidth, and achieves load balancing in P2P networks are not included in the scope of the paper. Instead, through theoretical proofs of the algorithm, its desirable features and its time complexity bound are given. Furthermore, the appropriateness of the algorithm for hybrid CDN-P2P networks is not considered. In the rest of the paper, we consider all these practical aspects of the algorithm and show its viability for hybrid CDN-P2P networks.

IV. NETWORK SIMULATION MODEL

In Section 3, we presented the system model assumed in [38]. In this section, we present a variation of the above model to make it practical for hybrid CDN-P2P networks. [16] merely

TABLE 1. Simulation model parameters.

Term	Description
n	Network dimension.
n-1	Number of neighbours = number of disjoint paths.
Peer id	Permutation of symbols 1, 2, 3, ..., n.
Routing Buffer	An implicit buffer for each peer used to hold the message shares to be routed on the path between the source peer and the destination peer.
lsp	last swap position
Routing Parameter	Peer id, lsp, and the destination id as explained in Section III-B
$\mathcal{T}$	Number of source peers.

provided an algorithm for routing between a single pair of peers over disjoint paths and some theoretical analysis along with the correctness proof of the algorithm. In contrast, the simulation in this paper carried out the presence of multiple sources and destinations allowing multiple concurrent message routing over all node-disjoint paths in a pipelined manner using PeerSim simulator for varying sizes and dimensions of star overlay graphs, demand, and buffer sizes per peer.

PeerSim [34] is an open source P2P systems simulator developed in Java at the Department of Computer Science, University of Bologna. It is designed as a scalable and dynamic simulator for large P2P networks as it aims to cope with P2P system properties and allows the user to replace its predefined entities by the user-entities. It supports two models of simulation: cycle -based and event -based, and can simulate both structured and unstructured overlays. In the cycle-based model, in each cycle, a peer is randomly selected and its protocol is executed. Whereas in the event -based model, nodal protocols are executed according to the message delivery time order [35]. Due to its scalability, support for cycle-driven simulation and star networks, accuracy, provisions for construction, execution, and data collection aspects of the simulation, we selected the PeerSim simulator.

The simulation proposed in this paper uses the star graph topology as in the inherently-stabilizing multipath routing algorithm for star P2P overlay networks introduced in [16]. We consider a star network consisting of a collection of peers that communicate through message exchange. Each peer is uniquely identified by an id, connected with its neighbours by bidirectional communication channels corresponding to edges in the star network, and runs the inherently-stabilizing multipath routing algorithm. The network is static; new peers cannot join a network, and existing peers may not leave or crash. Byzantine behaviour is not considered. Table 1 presents the model parameters related to the routing algorithm used in our simulation.

For the purpose of simulation, a cycle-driven model is assumed for the message routing, i.e., the simulation executes its steps in regular time intervals in which each step performed to complete the execution is referred to as a *cycle*. In each cycle of the simulation, each peer carries out the following two actions. First, if the peer's input buffer contains a message, the message is split into n-1 message shares where each message share is sent to a distinct neighbour. Second, each message share sent to a particular peer in the previous cycle is made available in the *routing buffer* of the recipient peer. Each peer maintains a routing buffer of a fixed size to store the received messages. In case a buffer element is not available, upon receipt of a message, message drop occurs. Then, each message share in the routing buffer is sent to the neighbour decided as per the routing parameters as descried in Table 1. Communication may incur unit time delays as a result of using the cycle-based simulation, and is not subject to any form of failures. No message shares may be lost; links between pairs of peers are always operational; and the integrity of messages is always maintained. Each system channel between two peers is assumed to have unit capacity and in the current cycle of the simulation, it can deliver a message share sent in the previous cycle.

In the beginning of each simulation cycle, a new set of input messages is randomly assigned to source peers to be sent to randomly selected destination peers, where each set of input messages consists of $\mathcal{T}$ messages. For each message, the destination peer is distinct from the source peer however a destination peer may be common for more than one source and a source peer may receive more than one message. In the first cycle of the simulation, each input message assigned to a source is split into n-1 message shares and each share is placed in the routing buffers of the source's neighbours as described by the algorithm. Subsequently, in the second cycle, while message shares are forwarded to other peers by the neighbours of the sources, a new set of input messages are assigned to new set of randomly selected source peers then distributed to their sources' neighbours, and so on. The rest of the steps for routing the messages is performed as described in Section III

In our simulation, this process is repeated in the first 21 cycles of the simulation where one new set of input messages are sent in each cycle. Therefore, a total of 21*$\mathcal{T}$ input messages are fed to the simulator. In 21 + (D (S_n) + 4) cycles, the routing of all the input messages is completed since the last set of input messages is added in the 21^{th} cycle of the simulation. Figure 2 summarizes the simulation process.

In our simulation, the largest diameter $D(S_n)$ of the networks we consider is 9. Therefore, it takes at most 13 cycles for each message to reach its destination. Recall that each message takes at most $D(S_n) + 4$ rounds to be routed.

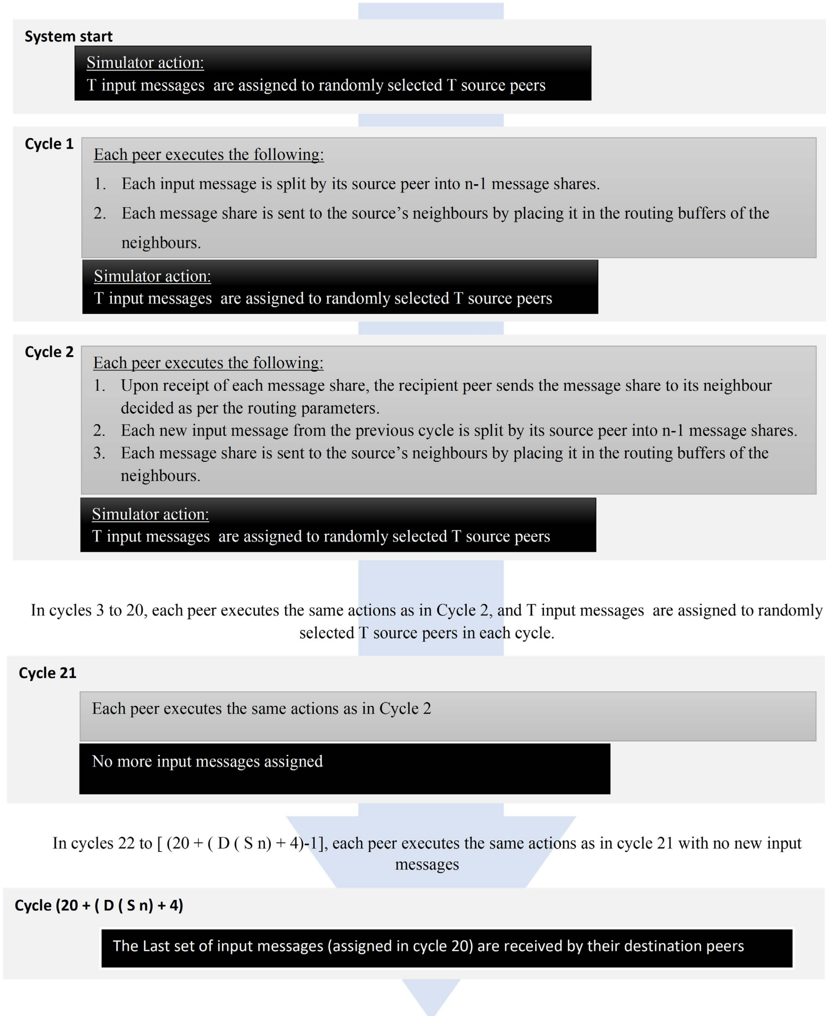

FIGURE 2. Simulation process.

Observe that in the first 13 rounds after the simulation starts, messages sent in a pipelined manner do not occupy all the multipath channels/processes provided that sufficiently many messages are sent in each cycle. On the other hand after the 13^{th} cycle, all/most channels and peers can be occupied by messages which show the real throughput capacity of the network. Therefore, we had to choose more than 13 cycles of simulation. We chose 21 cycles to experiment the network, to observe the network where peers and channels on parallel paths are fully or mostly occupied for sufficiently many cycles, 8 in this case.

Also observe that if sufficiently many, one or nearly one, messages is not sent in each cycle from each source, all channels and parallel paths cannot be kept busy to show the real throughput of the network. Therefore, we experimented with number of sources $\mathcal{T}$ between 2000 and 5000 where the maximum network size is 5040 which provide sufficient number of message to keep nearly all network channels busy.

Each performance evaluation experiment is simulated after repeating the simulation 20 times with dynamically and randomly selected source peers and destination peers. The average values of these repetitions are computed and shown, and individual simulation results for each experiment are shown whenever possible.

V. SIMULATION RESULTS

In this section, we present our simulation results related to message propagation delay, network throughput, buffer requirements, load balancing and scalability of the inherently-stabilizing multipath routing protocol for hybrid CDN-P2P networks. A cycle-based PeerSim simulator was used to evaluate these properties. To the best of our knowledge, no papers have used a cycle-based PeerSim simulation. PeerSim [34] is an open source P2P systems simulator developed in Java. To build a simulator, the user has to construct a network of peers; write protocols that represent the actions each peer will perform; choose a control to monitor the properties and modify the parameters of the network; run the simulation; then collect data.

A. NETWORK THROUGHPUT

In this section, we present the experimental results related to network throughput. Throughput is a fundamental service quality measure of CDN's due to being an important indicator of the quality of the network performance. The throughput of a network increases as the network load increases provided that channel capacities available across the network are exploited, network load is evenly distributed across network channels and peers, and network bottlenecks are eliminated.

In our simulation, we examined the effect of the network size and the number of source peers on the throughput. Therefore, we considered these two factors independently in two separated experiments. First, we observed the change in throughput as the network size is increased while the number of source peers is kept fixed for both single and multipath usage. Second, we varied the number of source peers and examined the change in throughput while the network size is kept fixed for both single and multipath usage. The throughput in the simulation is measured in bits per cycle and the buffer size for peers is of unlimited size for simulation purposes to avoid any message drop.

To measure system throughput for various network sizes, we used 5000 source peers and ran the simulation for networks of dimensions n = {4, 5, 6, 7}, where for each network size the simulation was repeated 20 times and average results were collected. Figure 3 shows the result of the simulation where the x-axis represents network size, and the y-axis represents the throughput measured for the associated network size for both single and multipath usage. In the figure, it can be seen that the throughput gradually increases as the network size increases. As seen in Figure 3, the throughput is increased by 314% for the single path routing and 331% for the multipath routing when changing the network form size 24 (n=4) to size 120 and increased by 410% for the single path routing and 481% for the multipath routing from network size 120 (n=5) to network size 720 (n=6). As mentioned in Section III, the size of an n-dimensional network is given by $n!$. For example, if $n = 5$, the network size is given by $5! = 120$. On the other hand, the ratio of increase in throughput between network size 720 to network size 5040 (n=7) is only 323% for the single path routing and 434% for the multipath routing. The significant improvement in throughput when the network size is increased is attributed to the following reasons. As the network size increases, the number of disjoint paths between peers in a star graph significantly increases which in turn improves throughput dramatically. On the other hand, as the network size increases, the number of available multipaths also increases which in turn increases the system throughput since additional paths can enlarge the communication bandwidth between communicating end points using the additional available disjoint paths. The throughput increase appears to be exponential with respect to the network dimension. This can be attributed to exponential increase in number of available disjoint paths between endpoints in star graphs. Observe that the throughput increase is slightly less for single path routing compared to that of multipath routing. This is attributed to the following. First, single path routing does not utilize all the available paths. Second, multipath routing leads to better load balancing and more congestion in some peers. Also observe that the throughput difference between multipath and single path routing widens as the network size grows. This is attributed to the avaliability of significantly more multipaths in larger networks that can not be exploited by single path routing. Hence, the star overlay networks have sufficiently many multipaths between all pairs of endpoints, utilizing the multipaths improves network throughput significantly.

To measure the system throughput for varying number of source peers, we used the network size of 5040 (dimension of n=7) and run the simulation for varying number of source peers of 500, 1000, 1500, …, 4000 as shown in Figure 4, where for each number of source peers the simulation was repeated 20 times and average results were collected. The x-axis in Figure 4 represents the number of source peers while the y-axis represents the throughput. It can be seen in Figure 4 that the network throughput increases linearly to the number of source peers for a fixed network size (5040). Our simulation results show that as the number of concurrent message transmissions (number of sources) increases, the amount of bits transferred per cycle also increases resulting in increased throughput. The increase in the throughput is achieved by the available network bandwidth between pairs of peers in star overlay networks provided by the large number of node-disjoint paths between them and load balancing of the content delivery in the network. Since we assumed unlimited buffer elements and no message drops are experienced as the throughput is increased in our experiments, we can conclude that the algorithm does not lead to bottlenecks and is scalable. We repeated the abovegiven experiment for single path routing and observed that multipath routing provides significantly better throughput regardless of the network size and the number of sources as shown in Figures 3 and 4. Figure 3 shows that as the network size increases, since the number of multipaths and the bandwidth increase, the throughput increases for the same demand. Figure 4 shows the throughput for various demand for the network size of 5040. It is easy to see that although the multipath routing yeilds significantly more throughput due to reducing congestion, as the number of sources (demand) increases, the throughput does not increase at the same rate for single path and multipath routing. This is attributed to reaching the level of congestion for both the routing schemes that does not allow the network bandwidth to be further increased. This result clearly establishes the viability of star overlay networks for hybrid CDN-P2P networks since the star overlay networks meet the significant bandwidth and throughput requirements of hybrid CDN-P2P networks.

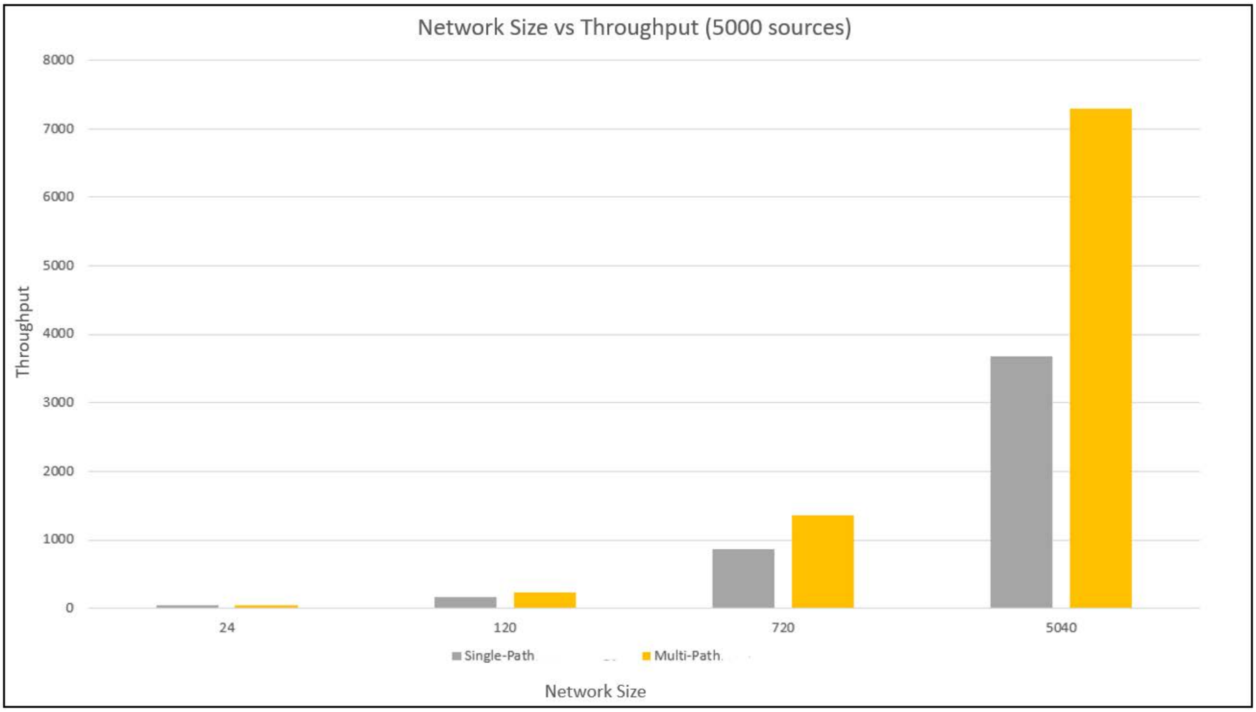

FIGURE 3. The throughput compared to the network size for a fixed number of source peers (5000).

Results given in Figures 3 and 4 show that multipath routing in star overlay networks provides significant network throughput for hybrid CDN-P2P networks.

B. BUFFER REQUIREMENTS

In this section, we estimate the buffer requirements for routing messages using the inherently-stabilizing routing algorithm [16]. In computer networks, a buffer is a physical memory used by the network components to temporarily store an amount of data while its being transferred from one component to another. In our simulation, we estimated the buffer requirements of network peers for varying network sizes, demand (number of source peers), and single path and multipath routing separately. We assume that each buffer element is capable of holding a single message share in the routing process.

First, the algorithm was simulated for each network dimensions of n = {3, 4, 5, 6, 7} to show the effect of the network size on the buffer size requirement when using multipath routing. For each network size, the algorithm was simulated while increasing the buffer sizes at each run until finding the minimum buffer size where the algorithm never experiences any message drops and the results are shown in Figure 5. In all simulations the number of source peers was fixed to $\mathcal{T} = 2000$. Through our initial simulations, we discovered that small scale networks require roughly on the order of ten times more buffer elements and when the buffer size is increased by 100, we are able to find the buffer requirements in a reasonably many simulation experiments. Similarly, we discovered that for large scale networks, when the buffer size is increased by 10, we are able to find the buffer size requirements in a reasonably many simulation experiments. Therefore, in each run, for small scale networks with $n<6$, the buffer is increased by 100, whereas, for large scale networks with $n\geq6$, the buffer size is increased by 10. As we increase the buffer sizes, we observe the effect of the buffer sizes on network throughput. When the buffer sizes are insufficient, expected throughput cannot be obtained due to message drops. However, when the buffers reach sufficient sizes, additional buffer size increases do not lead to throughput increases. Accordingly, at the end of each simulation run, the throughput is calculated for larger buffer sizes until the network no longer experience any message drops. The throughput is calculated only for the message shares that successfully reached the destination. The smallest buffer size to provide the maximum throughput is considered as the suitable buffer size. For example, for a network of size 5040 (n=7), we ran the simulation first using a buffer size of 10, then calculated the throughput at the end of the simulation using the message shares that successfully reached the destination. Then, we ran the simulation again using a buffer size of 20 and calculate the throughput. In the next simulation, we use a buffer size of 30, and so on until we obtain 5 simulations that have the same throughput and consider the minimum buffer size which no longer improves throughput as the required buffer size for the network of size 5040. It can be observed that for these simulations where the throughput no longer increases though the buffer sizes are increasing, the network does not experience any message drops.

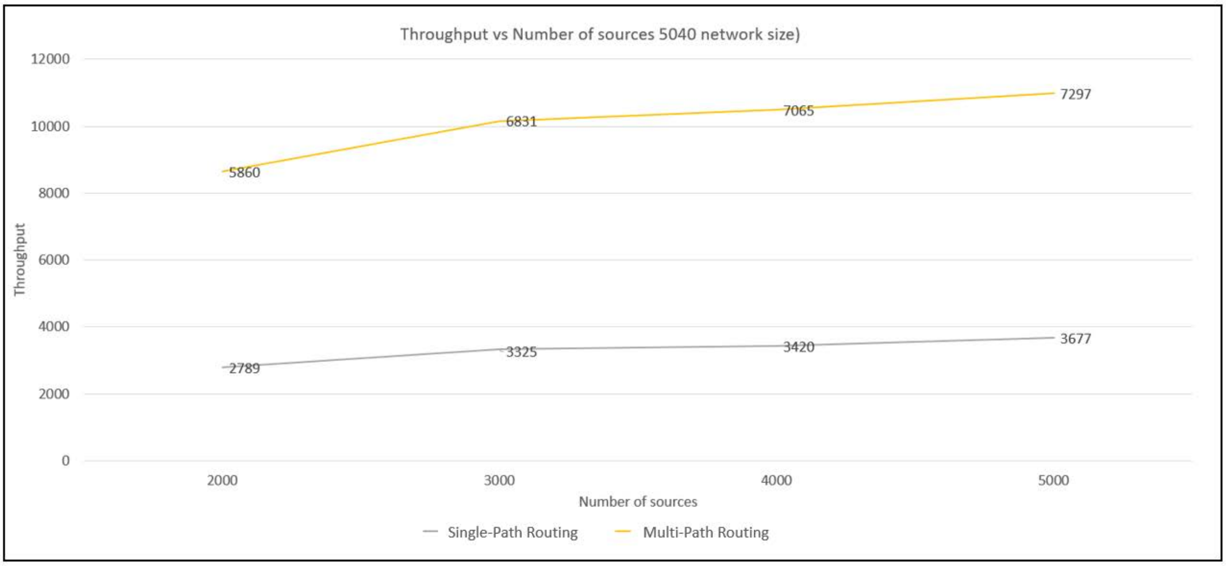

FIGURE 4. The throughput versus the number of source peers for a fixed network size of 5040.

As explained in Section IV, each simulation is fed with 21*$\mathcal{T}$ input messages. Therefore, a total of 21*2000 input messages are used in each simulation run. Figures 5 (a,b) present the results of our simulation where the x-axis represents the buffer sizes and the y-axis represents the network throughput. As shown in Figure 5 (b), for the network size of 720, the throughput linearly increases from buffer size 50 to 800 and once the buffer size of 800 is reached, the throughput remains the same since no message drop takes place. Hence, the suitable buffer size for a network of size of 720 with 2000 sources is found to be 800. Figure 5 also shows that when sufficient buffers are available, the system throughput cannot be increased beyond a certain point for each network size. This is due to the full utilization of all available multipaths and the unavailability of additional multipaths to increase the throughput further. It can be observed that when the network size is increased, more multipaths become available and the network throughput increases.

It can be observed from Figure 5 that the increase in buffer and network sizes increases system throughput. In addition, Figure 5 clearly shows that when the network size increases, the buffer requirements decrease for the same number of sources. This is due to the routing of less number of messages per peer in the routing process. This also shows that load balancing is achieved by the algorithm. It can be seen that as we increase the buffer size, the throughput of a network increases until it becomes stable at some point.

The simulations to obtain Figure 5 are repeated using various number of source peers $\mathcal{T} = \{2000, 3000, 4000, 5000\}$ and the results are shown in Figure 6. Figure 6 depicts the effect of the network size on the peers buffer sizes where the x-axis represents the network size while the y-axis represents the buffer size. The buffers sizes shown are the buffer sizes that do not cause any message drop for the network size under consideration and obtained through repeated experimentation where buffer sizes are gradually increased to find the sufficient buffer size to prevent message drops. It can be concluded from the graph that for a fixed number of source peers, the buffer size linearly decreases as the network size increases. It is observed that as the network size increases, more peers are involved in the routing of input messages, therefore a reduced buffer size is required as the number of message shares routed per peer reduces. It can be seen that for any network size greater than 720 as long as we are using a buffer of size 800, the algorithm does not experience message drops.

Second, to show the effect of the number of sources on the buffer size required for each peer, the algorithm was simulated using different number of source peers $\mathcal{T} = \{2000, 3000, 4000, 5000\}$ while keeping the same network size. For each number of sources, the simulation was run while increasing buffer sizes for each run until reaching the suitable buffer size which causes no message drops. These simulation steps were repeated for different network sizes and the results are shown in Figure 7.

In Figure 7, the x-axis represents the number of source peers while y-axis represents the buffer size. It can be seen that for a fixed network size, the buffer size is linearly increasing as the number of source peers increases. It can clearly be seen that for a sufficiently large network size (5000 peers), very small, nearly constant, buffer sizes are sufficient even in the presence of high demand (large number of sources in our experiment). This verifies the viability of star overlay networks for hybrid CDN-P2P network in terms of buffer requirements under heavy demand.

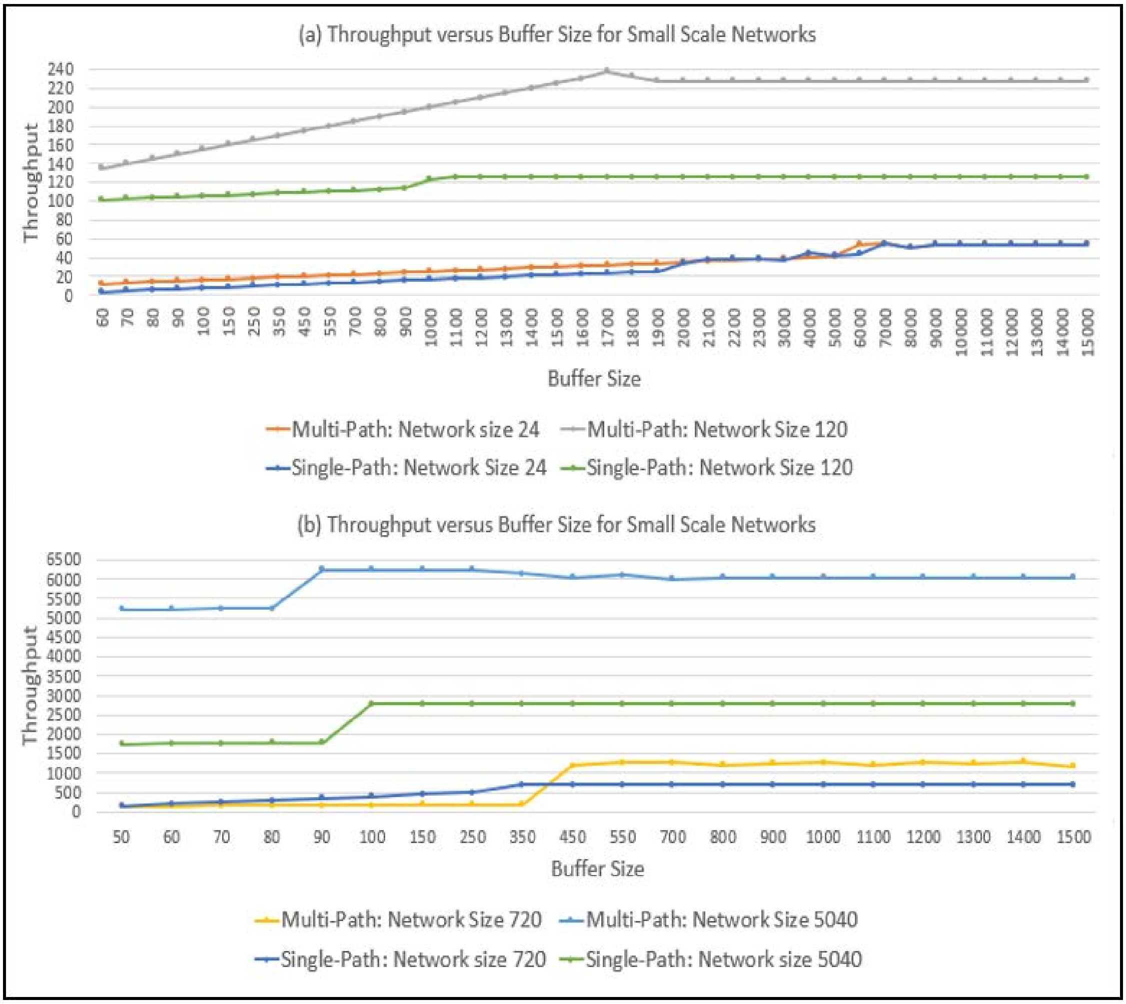

FIGURE 5. **Throughput versus buffer size and number of sources=2000.**

From Figures 6 and 7, it can also be seen that for small scale networks of n=4 or 5, as the number of source peers (the number of input messages) increases, each peer requires larger buffers in order to avoid message drops. The required buffer size for the small scale networks is 95% larger than the size required for large scale networks.

In the figures, it can clearly be seen that when the network and the buffer sizes are increased, since the number of multipaths is increased and message drops decreases, system throughput is increased. It can be observed that when the network size is increased for the same demand and the buffer size, message drops decrease as shown in Figure 5. From this observation, it can be concluded that the algorithm achieves a high degree of load balancing leading to reduced buffer requirements for larger network sizes for the same demand for single path routing. It is easy to see in Figure 9 that the buffer requirements are more for multipath routing than those of the single path for varying network sizes and demand for the same reason as discussed earlier.

Figure 5 captures the effect of the buffer size on network throughput where due to insufficient buffer size message drops occur for smaller buffer sizes. As a result, maximum throughput cannot be obtained. However, when buffer sizes are increased to a level where they are sufficient, additional buffer size increases do not lead to throughput increases. It can be seen that for network size 720, the throughput linearly increases from buffer size 50 to 800 and once the buffer size of 800 is reached, the throughput remains the same since no message drop takes place. Hence, the suitable buffer size for a network of size 720 with 2000 sources is found to be 800.

Figure 6, shows the buffer requirements for various network sizes and 5000 sources and for both single path routing and multipath routing. The buffer requirements for multipath routing is slightly more than that of single path routing since the throughput for multipath routing is significantly more and it is natural that when more messages are routed per cycle, the buffer requirements increase. Figure 7 shows the buffer

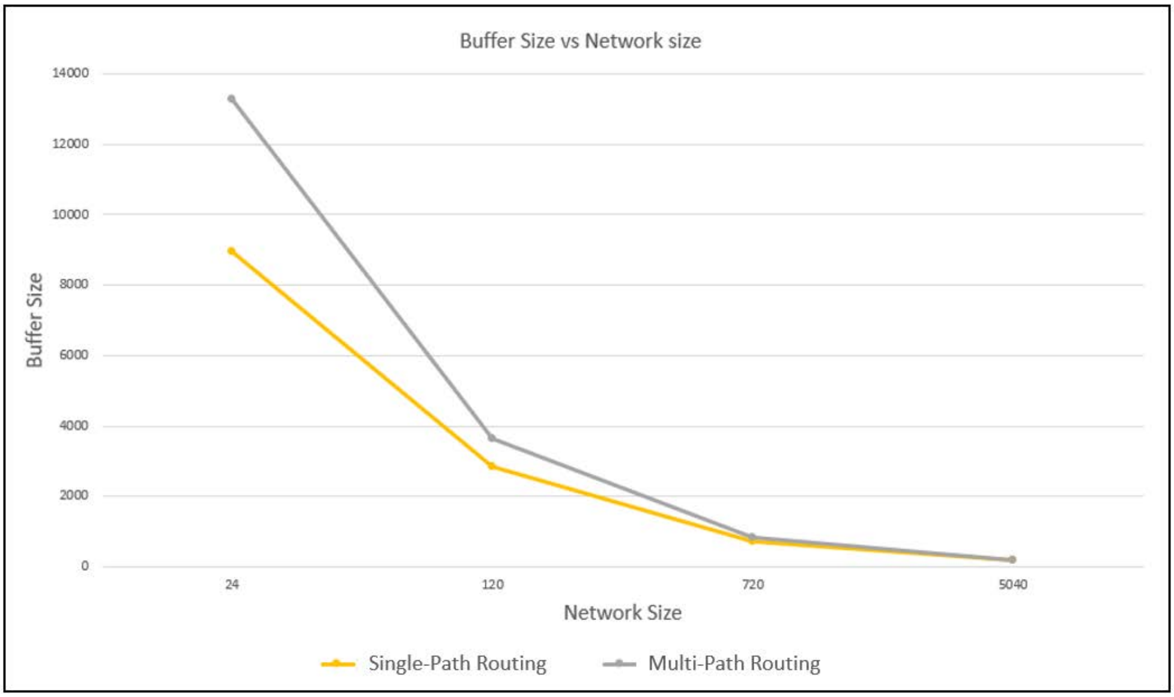

FIGURE 6. Network size versus buffer size and number of sources=5000.

requirements for various number of sources and various network sizes for single path and multipath routing. Observe that for smaller networks (sizes of 24, 120 and 720) the buffer requirements increases as the number of sources increase. On the other hand, for larger network sizes (such as 5040), the buffer requirements increase only very slightly. This is attributed to a decrease in network congestion as the network size grows for the same network size.

C. LOAD BALANCING AND SCALABILITY

Load balancing is desirable in CDN's as distributing the network load among various CDN components improves the resource utilization and the response time while eliminating network bottlenecks. Load balancing also helps avoiding heavy load in some network components while others are idle or have significantly less load. Therefore, a good distribution of the network load means a faster response to the end users requests. Many modern applications such as online gaming, video streaming, and etc. often generate heavy network traffic that cannot run without proper load balancing. In this section, we experimentally show that multipath routing in star overlay networks [16] achieves load balancing for hybrid CDN-P2P networks.

The simulation was carried out on a network with 5040 peers (n=7) with 2000 source peers. A total of 21*2000 input messages were sent during the simulation. To evaluate the distribution of the load among the network peers, we count the number of times each peer is traversed by a message share. Based on the results shown in Figure 5, the buffer size used in this simulation is chosen as 90 which is the suitable buffer size for a network of size 5040 with 2000 source peers.

Figures 7 and 10 show that as the network size is increased, the buffer requirements dramatically reduce for the same demand. This clearly shows that the algorithm achieves a high degree of load balancing by distributing messages among more peers as the network size grows.

Figure 10 shows the load balancing of the network peers by depicting the number of times each peer is visited to complete all the message transmissions. The x-axis of the graph denotes the number of peers in the network while the y-axis shows visit frequency of each peer to show the distribution of the load. It can be concluded from Figure 10 that the visit frequency of most of the peers are close to the average visit frequencies with a small standard deviation of 20.7 for fixed demand. Figure 10 also shows that the degree of load balancing remains the same regardless of the network size for the same demand. Therefore, clearly the load in the network is fairly evenly distributed among all the network peers in a similar manner for various network sizes. Thus, the multipath routing for star overlay networks is experimentally shown to provide balanced load distribution.

In addition, since the algorithm increases the degree of load balancing and decreases buffer requirements as the network size grows, it is highly scalable. It can be observed that as the demand is increased, buffer requirements increase for small networks of size of 24 to 120, and remains nearly the same for network sizes of 720 and larger. Notice that buffer size of 800 is sufficient for networks of size 720 and less, whereas, buffer size of less than 100 is sufficient for network of

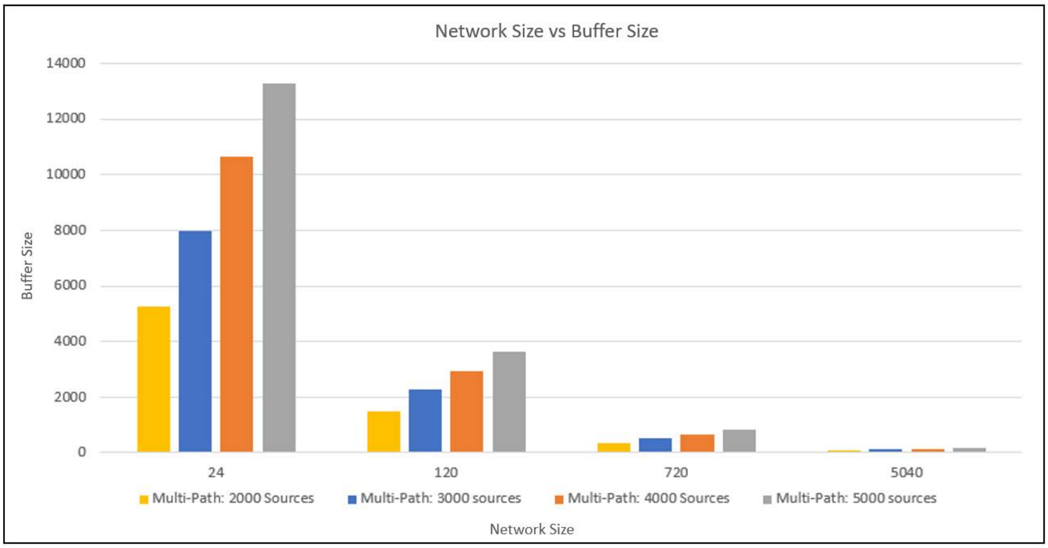

FIGURE 7. **Buffer size versus the network size.**

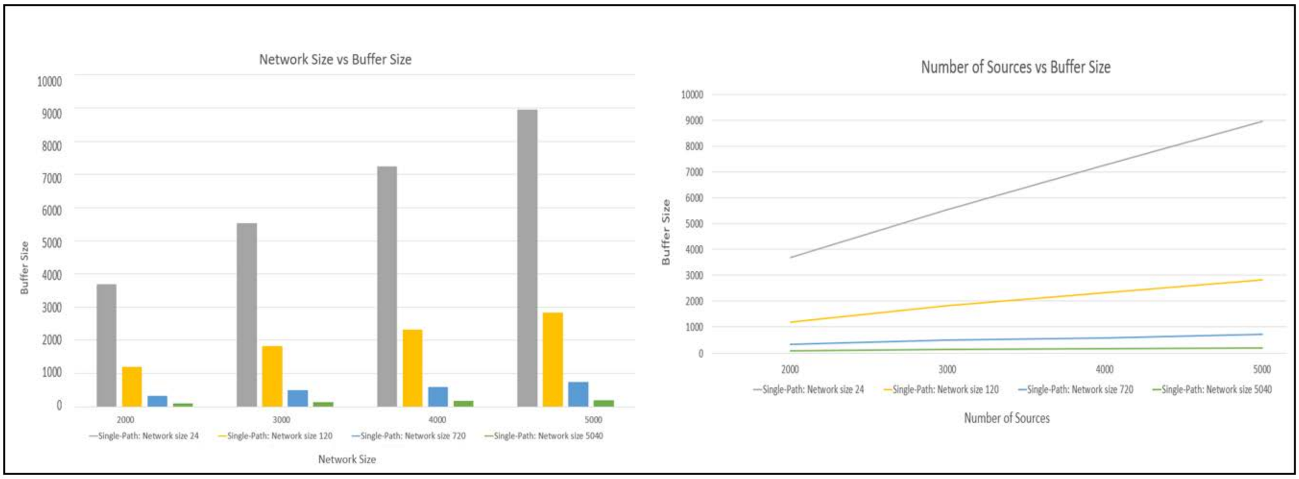

FIGURE 8. **Buffer size versus the network size.**

size 5040. This clearly shows that the algorithm achieves high degree of load balancing and scalability.

As shown in Figure 9 capturing the relationship between the buffer size and the number of source peers, as we increase the number of source peers, the buffer size only requires a slight increase. In addition, recall that for the network size of 5040, when handling 21*2000 messages, the required buffer size was only 800 and the buffer requirement increases only marginally when the demand (number of sources) is increased in a large network. Also, it can clearly be seen that for a large size network (with 5040 peers), very small size buffers are sufficient to eliminate message drops.

It is easy to observe from Figure 8 and 9, the buffer requirements are slightly less for multipath routing compared to single path routing although multipath routing yields a significantly more throughput compared to single path routing. This clearly shows that multipath routing yields to better load balancing. In addition, Figure 10 and 11 provide the distribution of node visit counts for various network sizes for single and multipath routing. The figures clearly show that multipath routing yeilds significantly better load distribution among nodes for all network sizes. Thus, we conclude that the multipath routing in star overlay networks for hybrid CDN-P2P networks is highly scalable since peers with existing buffers continue to route message without message drops when network size is increased.

D. MESSAGE PROPAGATION DELAY

In this experiment, we evaluate the propagation delay for a message to be transmitted from a source to a destination peer. In a hybrid CDN-P2P network, the message propagation delay is a major obstacle in the development as it affects the

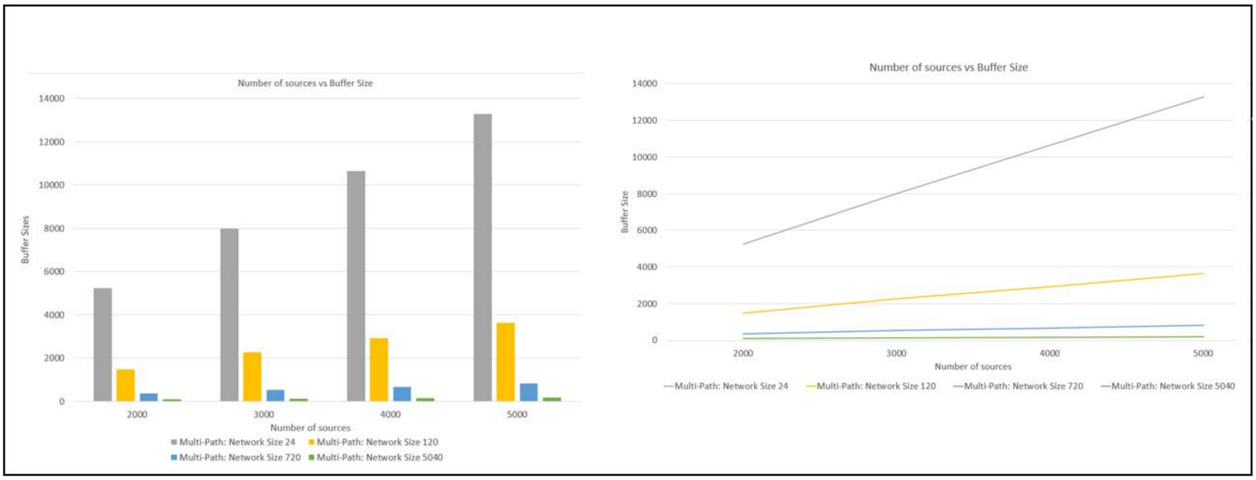

FIGURE 9. Required buffer size versus the number of source peers.

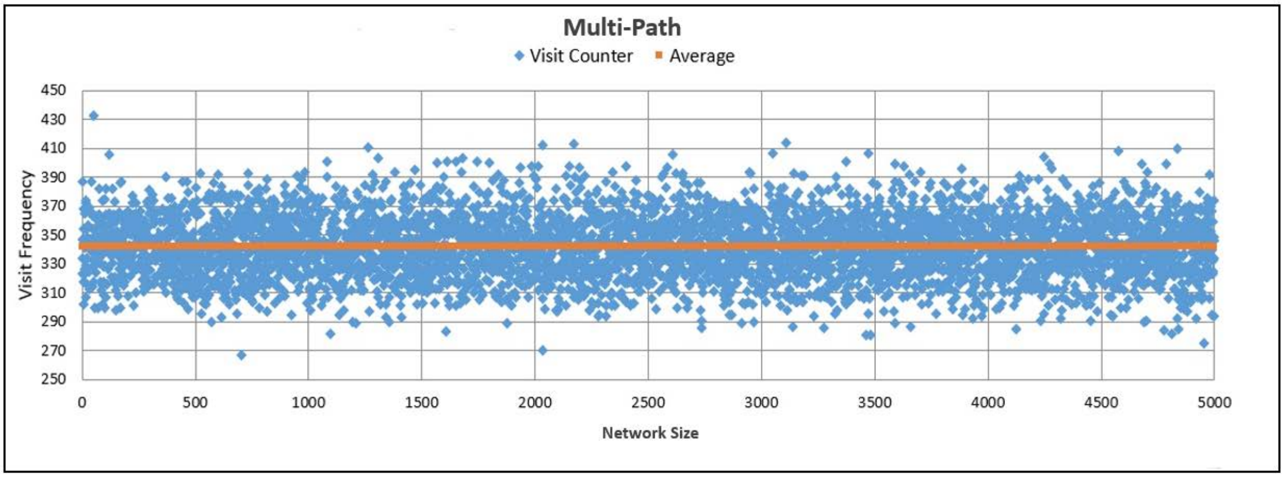

FIGURE 10. Load distribution of 4000 messages in a network of size up to 5040.

quality of service. The inherently-stabilizing routing algorithm proposed in [16] is theoretically proven that after the system start, the algorithm successfully delivers messages from source peers to destination peers in at most $D(S_n) + 4$ rounds/cycles where $D(S_n)$ denotes the diameter of the n-dimensional star network S_n. In order to experimentally show the correctness of the algorithm and that it requires $D(S_n)+4$ rounds (cycles in the simulation) to complete message delivery, we conducted simulation experiments. In particular, we observed the effect of the network size on the number of rounds/cycles required to complete the message propagation between a source peer and a destination peer. The simulation was run for various network dimensions ($n = \{3, 4, 5, 6, 7\}$) and the total number of cycles required to complete the message transmission was measured at the end of each run. The experiments are repeated 20 times for each dimension and the average number of cycles in the 20 experiments is taken as the number of cycles required for the dimension under consideration to be depicted. The results of our simulation experiment is shown in Figure 12, where the x-axis denotes the network size and the y-axis denotes the number of rounds. Observe that the number of rounds/cycles required to complete the message transmission increases slightly with the network size. This stems from the fact that the round complexity has to do with the diameter of the graph and the diameter increases slightly as the network size increases. The same experiment also verifies the correctness of the multipath routing algorithm proposed in [16]

In order to verify the correctness of the theoretically proven number of round/cycles of $D(S_n) + 4$, we compared the simulation results and the theoretically calculated values. The theoretical values were computed using the equation of $D(S_n) + 4$ on the network dimensions of $n = \{3, 4, 5, 6, 7\}$; where $D(S_n) = \lfloor 3(n-1)/2 \rfloor$. For instance, for a network of

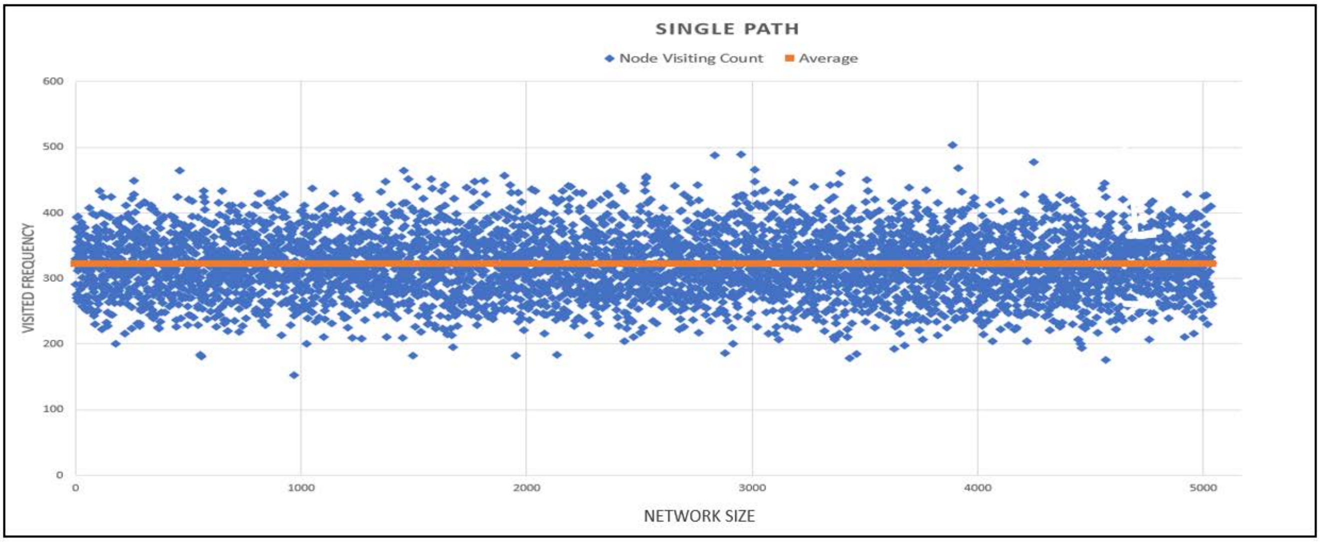

FIGURE 11. Single Path Routing: Load distribution of 4000 messages in a network of size up to 5040.

dimension n=3, $D(S_n) = \lfloor ^{3(3-1)}/_2 \rfloor = 3$. Therefore, the algorithm successfully delivers messages from source peers to destination peers in 3+4 =7 rounds/cycles. The comparison between the simulated and theoretically calculated number of rounds/cycles required for a message transition are shown in Figure 12. The results shown verifies the correctness of the theoretically found round complexity (number of cycles) as the actual number of rounds/cycles obtained in the simulation are very close to the theoretically calculated rounds/cycles for varying dimensions of the star graphs. For example, for a network of dimension n=3, the theoretical number of rounds is 7 while in the simulation it is shown to require 4 rounds/cycles. That is, based on Figure 12, we showed by simulation results that the algorithm successfully completes the message delivery in at most $D(S_n) + 4$ rounds.

E. SUMMARY AND DISCUSSIONS

In this section, we summarize the empirical evaluation of the inherently-stabilizing multipath routing protocol for non-traditional overlay nextworks (star networks) under a realistic model for hybrid CDN-P2P networks. In particular, we empirically show that the claimed yet unproven properties such as load balancing, buffer requirements, scalability, and throughput of the multipath routing algorithm of Karaata and Alsulaiman [16]. Reference [16] merely provides an algorithm for routing between a single pair of peers over disjoint paths and some theoretical analysis along with the correctness proof of the algorithm. In contrast, the simulation in this paper has been carried out in the presence of multiple sources and destinations allowing multiple concurrent message routings over all node-disjoint paths in a pipelined manner using PeerSim simulator for varying sizes and dimensions of star overlay graphs, demand, and buffer sizes per peer.

Simulation results show that the growth in the size of the network slightly increases the throughput achieved by the algorithm. Since when the dimension of the star overlay networks increases, sufficiently many multipaths become available between all pairs of endpoints whose effective utilization by the algorithm allows network throughput to increase significantly. In addition, our simulation results show that as the demand increases, the network throughput also increases though the network size is kept the same. The increase in the throughput is facilitated by the algorithm through utilizing the available network bandwidth between pairs of peers in star overlay networks provided by the large number of node-disjoint paths between them and load balancing of the content delivery in the network. Observe that for an increased demand of content routing for the same network size, throughput could not have been increased linearly without load balancing of the content routing among peers. Therefore, linear throughput increase for increasing demand for the same network size showed that during the distribution of the message shares from multiple sources to multiple destinations, the load on the network was fairly evenly distributed among all the peers by the routing algorithm thus achieving load balancing in the network. It can clearly be seen that the load balancing of a large number of content by the routing algorithm avoids the formation of bottlenecks in the scalable networks. Moreover, it can be observed that increases in the buffer and network sizes increase system throughput. In addition, as the network size is increased, the buffer requirements dramatically reduce for the same demand. The decrease in the buffer sizes as the network size increases clearly shows the effectiveness of the algorithm in load balancing via the use of available additional multipaths. In a separate experimentation, it is shown that the degree of load balancing remains the same regardless of the network size for the same demand. In addition, since the algorithm increases load balancing and decreases buffer requirements as the network size grows, the scalability of the multipath routing algorithm is established. The routing buffer size requirements of the algorithm for each

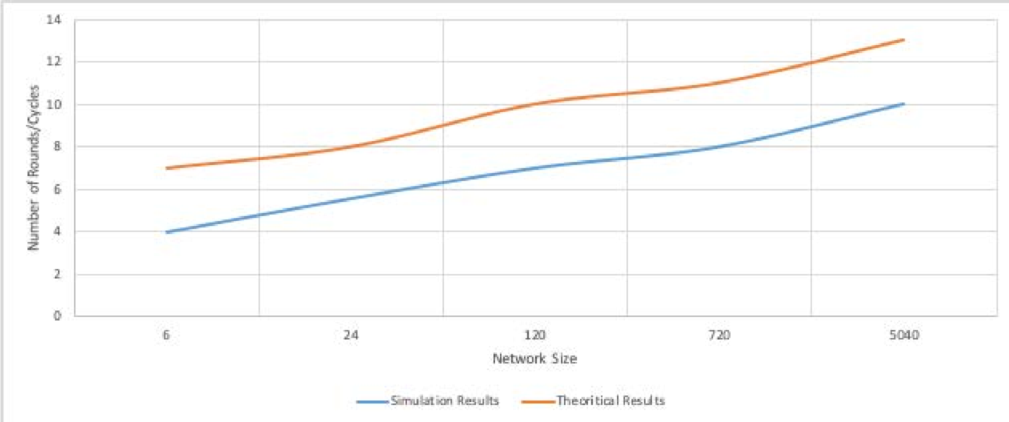

FIGURE 12. The number of cycles to complete message delivery versus network size.

peer is analyzed by monitoring the effect of the network size and the number of source peers on the buffer size. Simulation results also show that multipath routing improves the network throughput and the degree of load balancing, and reduce the buffer requirements compared to single path routing. The results obtained clearly establish viability of the multipath routing in star overlay networks.

Observe that better simulation results are not obtained since when the simulation is started, most routing buffers are empty and most channels are idle and they remain the same for a while until the message shares on these paths populate the network in a pipelined manner and maybe load is not well distributed.

VI. CONCLUSION

In this paper, we show that multipath routing in star overlay networks facilitates a high degree of load balancing, throughput enhancement, reduces buffer requirements and network bottlenecks and scalability. As these algorithmic properties are highly desirable for hybrid CDN-P2P networks, we establish the viability of the star overlay networks as an edge network for hybrid CDN-P2P networks to meet their content delivery quality of service requirements.

We anticipate that this work encourages researchers to consider overlay networks with abundant multipaths as edge networks for hybrid CDN-P2P and other networks. We also expect researchers to investigate other algorithmic properties obtained through the use of multipaths that are not considered here.

Although the obtained results are highly promising, better results can be obtained using higher demand or cycles. Our work only consider the benefits of the star overlay networks, however, the limitations by the underlying physical network that the overlay is mapped to is out of the scope of the current work.

In this work, we only considered point-to-point multicast communication. It is an open problem to apply multipath routing to other forms of communications including one to many and many to many. In this work, we only consider star overlay networks that provide a larger number of multipaths between any two endpoints. As future work, other overlay networks such as hypercubes can be considered and compared against star overlay networks with respect to the algorithm considered. It is also an open problem to enhance existing commercial hybrid CDN-P2P network applications using the multipath routing over star overlay networks.

The simulation conducted does not consider the effects of varying message sizes and the routing delays at peers. First, the routing mechanism described is based on local knowledge and is fairly simple, therefore the negligible delay caused by peers to identify peers to forward messages is not considered. Second, as it can be seen in Figure 6, a star overlay network provides significant bandwidth through the use of multipaths. When the message size is doubled, since we assume that the message size is of maximum of the capacities of all multipaths between two endpoints, the additional message size is accommodated in the next cycle and therefore takes only one additional cycle provided that another message does not arrive at the same source in the next cycle. Since the delay caused by increased message sizes is relatively simple to calculate as discussed above, additional experiments were not conducted for that purpose. As mentioned in Page 2, Paragraph 2, the role of CDN is that when a particular content is not available in the P2P network, a CDN component acts as a source peer in the edge network and provides the content. Since, a CDN component is viewed as a source peer and it assumes the role of source peer, its role is not considered separately.

In our current work, we only considered an abstract model of simulation where varying capacities and delays of communication channel are not considered. In addition, we did not consider the mapping of the star overlay network to the underlying physical network and the limitations brought by the mapping. An emulation of the proposed algorithm in a real network considering all the above is highly involved and out of the scope of the current work. We consider such an emulation as future work.

ACKNOWLEDGMENT

The authors are would like to thank the anonymous referees for their suggestions and constructive comments on an earlier version of the paper. Their suggestions have greatly enhanced the readability of the paper. The authors would also like to thank Aisha Dabees for her assistance on algorithm simulation and paper revision.

REFERENCES

[1] G. Tang, H. Wang, K. Wu, and D. Guo, "Tapping the knowledge of dynamic traffic demands for optimal CDN design," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 98–111, Feb. 2018.

[2] K. Srinivasan, A. Mubarakali, A. S. Alqahtani, and A. Dinesh Kumar, "A survey on the impact of DDoS attacks in cloud computing: Prevention, detection and mitigation techniques," in *Intelligent Communication Technologies and Virtual Mobile Networks*, S. Balaji, Á. Rocha, and Y.-N. Chung, Eds. Cham, Switzerland: Springer, 2020, pp. 252–270.

[3] M. Day, B. Cain, G. Tomilson, and P. Rzewski, *A Model for Content Internetworking (CDI)*, RFC Editor document RFC3466, Feb. 2003. [Online]. Available: https://tools.ietf.org/html/rfc3466

[4] H. Yin, X. Liu, G. Min, and C. Lin, "Content delivery networks: A bridge between emerging applications and future IP networks," *IEEE Netw.*, vol. 24, no. 4, pp. 52–56, Jul./Aug. 2010.

[5] Y. Kim, Y. Kim, H. Yoon, and I. Yeom, "Peer-assisted multimedia delivery using periodic multicast," *Inf. Sci.*, vol. 298, pp. 425–446, Mar. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514011141

[6] H. B. Hadj Abdallah and W. Louati, "Ftree-CDN: Hybrid CDN and P2P architecture for efficient content distribution," in *Proc. 27th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Feb. 2019, pp. 438–445.

[7] R. Peterson and E. G. Sirer, "AntFarm: Efficient content distribution with managed swarms," in *Proc. NSDI*, 2009, vol. 9, no. 1, pp. 107–122.

[8] R. Huang and Y. Zhang, "Survey of WEBRTC based P2P streaming," IETF, Fremont, CA, USA, Tech. Rep., Feb. 2014, pp. 1–11. [Online]. Available: https://www.ietf.org/archive/id/draft

[9] (2020). *Discover CDN Mesh Delivery for Live Streaming and Videoon-Demand*. Accessed: May 5, 2020. [Online]. Available: https://streamroot.io/cdn-mesh-delivery/

[10] *Peer5*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.peer5.com/cdn

[11] S. Nacakli and A. M. Tekalp, "Controlling P2P-CDN live streaming services at SDN-enabled multi-access edge datacenters," *IEEE Trans. Multimedia*, vol. 23, pp. 3805–3816, 2021.

[12] T.-W. Um, G. M. Lee, and H.-W. Lee, "Trustworthiness management in sharing CDN infrastructure," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 73–75.

[13] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 2001–2014, Dec. 2013.

[14] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "A case for a coordinated internet video control plane," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 359–370, 2012.

[15] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 362–373, Aug. 2011, doi: 10.1145/2043164.2018478.

[16] M. H. Karaata and T. Alsulaiman, "An optimal inherently stabilizing routing algorithm for star P2P overlay networks," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 17, pp. 4405–4428, Dec. 2016.

[17] S. Fujita, "A new network topology for P2P overlay based on a contracted star graph," in *Proc. 10th Int. Symp. Pervasive Syst., Algorithms, Netw.*, Washington, DC, USA, Dec. 2009, pp. 29–33, doi: 10.1109/I-SPAN.2009.51.

[18] C.-S. Lin and I.-T. Lee, "Applying multiple description coding to enhance the streaming scalability on CDN-P2P network," *Int. J. Commun. Syst.*, vol. 23, no. 5, pp. 553–568, 2010.

[19] Z. H. Lu, X. H. Gao, S. J. Huang, and Y. Huang, "Scalable and reliable live streaming service through coordinating CDN and P2P," in *Proc. IEEE 17th Int. Conf. Parallel Distrib. Syst.*, Dec. 2011, pp. 581–588.

[20] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Commun. ACM*, vol. 49, no. 1, pp. 101–106, 2006.

[21] S. Nakamoto. (Mar. 2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://metzdowd.com

[22] V. Buterin. (2013). *Ethereum Whitepaper*. [Online]. Available: https://ethereum.org/whitepaper/

[23] J. L. Ferrer and E. M. Payeras, "Cryptocurrency P2P networks: A comparison analysis," Menorca, Illes Balears, Spain, Tech. Rep., 2016, pp. 423–428.

[24] *Dalesa*, 2010. [Online]. Available: http://www.dalesa.lk/

[25] W. Garbe. (2005). *FAROO*. [Online]. Available: http://www.faroo.com

[26] Y. Psaras and D. Dias, "The interplanetary file system and thefilecoin network," in *Proc. 50th Annu. IEEE-IFIP Int. Conf. Dependable Syst. Netw.-Supplemental Volume (DSN-S)*, 2020, pp. 80–80.

[27] A. Ali, M. Khan, M. Saddique, U. Pirzada, M. Zohaib, I. Ahmad,and N. Debnath, "Tor vs I2P: A comparative study," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Kansas City, MO, USA, 2016, pp. 1748–1751.

[28] Y. Ye, B. Lee, R. Flynn, N. Murray, and Y. Qiao, "HLAF: heterogeneous-latency adaptive forwarding strategy for peer-assisted video streaming in NDN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 657–662.

[29] H. Dai, J. Lu, Y. Wang, and B. Liu, "A two-layer intra-domain routing scheme for named data networking," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 2815–2820.

[30] E. Kalogeiton, T. Kolonko, and T. Braun, "A multihop and multipath routing protocol using NDN for VANETs," in *Proc. 16th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2017, pp. 1–8.

[31] C.-H. Lee, J.-X. Lin, J.-S. Jhou, and K.-J. Chang, "Preliminary study on the multi-path pre-caching mechanism for content delivery network," in *Proc. IEEE 6th Global Conf. Consum. Electron. (GCCE)*, Oct. 2017, pp. 1–2.

[32] V. Poliakov, L. Sassatelli, and D. Saucez, "Adaptive video streamingand multipath: Can less be more?" in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018.

[33] S. Sur and P. K. Srimani, "Topological properties of star graphs," *Comput. Math. Appl.*, vol. 25, no. 12, pp. 87–98, Jun. 1993.

[34] (2005). *Peersim: A Peer to Peer Simulator*. [Online]. Available: http://peersim.sourceforge.net/

[35] B. L. Stephen Naicken, A. Basu, and S. Rodhetbhai, "A survey of peer-to-peer network simulators," in *Proc. 7th Annu. Postgraduate Symp.*, 2006, pp. 1–8.

MEHMET KARAATA received the B.S. degree in computer science from the University of Hacettepe, Ankara, Turkey, in 1987, and the M.S. and Ph.D. degrees in computer science from The University of Iowa, Iowa City, USA, in 1990 and 1995, respectively. He joined Bilkent University, Ankara, as an Assistant Professor, in 1995. He is currently working as a Professor with the Department of Computer Engineering, Kuwait University. His research interests include mobile computing, distributed systems, fault-tolerant computing, and self stabilization. He has earned the Distinguished Best Young Researcher Award and a Researcher Award from Kuwait University, Kuwait, in 2001 and 2009, respectively.

ANWAR AL-MUTAIRI received the B.S. and M.S. degrees from the Computer Engineering Department, Kuwait University, in 2013 and 2016, respectively. She is currently working as a Computer Engineer with the Center of Information Systems, Kuwait University. She expects to start her Ph.D. degree in the near future. Her research interest includes distributed algorithm development and verification.

SHOUQ ALSUBAIHI received the B.S. and M.Sc. degrees in computer engineering from Kuwait University, Kuwait, in 2005 and 2008, respectively, and the Ph.D. degree in computer engineering from the University of California, Irvine, USA. She is currently an Assistant Professor with the Computer Engineering Department, Kuwait University. Her research interests include distributed systems, parallel computing, design automation, evolutionary computation, and fault tolerance.