IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 15, 2017, accepted November 17, 2017, date of publication November 29, 2017,
date of current version December 22, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2778339

Novel Cooperative and Fully-Distributed Congestion Control Mechanism for Content Centric Networking

ANSELME NDIKUMANA[1], SAEED ULLAH[1], KYI THAR[1], NGUYEN H. TRAN[1], (Member, IEEE), BANG JU PARK[2], AND CHOONG SEON HONG[1], (Senior Member, IEEE)
[1]Department of Computer Science and Engineering, Kyung Hee University, Gyeonggi-do 17104, South Korea
[2]Department of Electronic Engineering, Gachon University, Gyeonggi-do 13120, South Korea

Corresponding author: Choong Seon Hong (cshong@khu.ac.kr)

This work was supported by the Institute for Information and Communications Technology Promotion grant through the Korea Government (Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices) under Grant 2015-0-00557.

ABSTRACT The router's buffer accommodates transient packets to guarantee that the network's links do not become idle. However, buffer overflow causes packet loss, which is a signal of congestion. In content centric networking (CCN), the interest packet, which is used for requesting content, may be dropped due to such congestion. Each interest packet is assigned a specific lifetime, and when the lifetime expires without obtaining the requested content, the consumer needs to resend the Interest. However, waiting for the expiration of an Interest's lifetime for retransmission is only appropriate for best effort traffic rather than services that are delay sensitive. In order to provide delay-sensitive applications with better quality of service, we propose a congestion control mechanism for CCN, in which, we prevent congestion before it happens through monitoring buffer size. Upon reaching the buffer threshold, the node notifies its downstream node. On receiving the notification, the downstream node adjusts traffic rate by allocating new incoming Interests to other face(s). However, when the downstream node fails to reduce traffic rate, the same procedure continues until the consumer node reduces sending rate. The simulation results show that the proposed mechanism is capable of significant performance improvements, with higher throughput.

INDEX TERMS Content centric networking (CCN), congestion control mechanism, cooperative and memory-efficient token bucket (CMTB), fully-distributed congestion control (FDCC), reduce sending rate (RSR).

I. INTRODUCTION

A. BACKGROUND AND MOTIVATIONS

In the past few decades, the Internet has expanded to an enormous extent, and this growth is expected to continue in the coming years. In accordance with this growth, the demand for delay-sensitive applications is also increasing [1]. According to the CISCO Visual Networking Index (VNI), in 2020, global Internet traffic will reach 95 times the volume of global Internet traffic in 2005, and the video traffic will be 82% of all consumer Internet Traffic [2]. To deal with this increasing demand, a clean-slate Internet architecture, namely Information Centric Networking (ICN) [3] has been introduced, which has transformed the concept of the Internet from end-to-end to content-specific communication.

Content Centric Networking (CCN) is one of the most promising ICN architectures, which considers content to be primitive in the communicative process, where content is requested and retrieved by name rather than by the specific IP address [4]. In CCN, there are two types of packets: the Interest packet, which is used for requesting content, and the Data packet, also called Chunk, which is transmitted in response to the Interest packet in the reverse path of Interest packet. In the subsequent analysis, unless otherwise stated, we will refer to Data packet and Chunk interchangeably.

For Interest and Data packets forwarding, CCN uses three main data structures: (1) the Content Store (CS), (2) the Pending Interest Table (PIT), (3) and the Forwarding Information Base (FIB). To get content, consumer sends Interest packet. When a router receives the Interest, it checks whether the requested content is cached in its CS. If the requested content is available in the CS, the router returns the Data packet to the consumer in the reverse path of the Interest packet. If the content is not cached in the CS, the router checks in PIT, which contains records of Interests that have been forwarded upstream and the faces/interfaces from which the Interests are received from. In case of the Interest packet for the same content has been forwarded before, the node appends the face on which the Interest arrives to the list of faces of matched PIT entry. In the absence of any matching PIT entry, a new entry is created in the PIT. The router checks in the FIB, which has outgoing faces, to determine where to forward the Interest packet [5].

In CCN, each Interest packet has a specific lifetime, and when the lifetime expires without obtaining the requested Data object, the Interest is removed from the PIT, and the consumer needs to retransmit the Interest packet [6]. Interest lifetime may expire due to congestion, or its duration is shorter than the network delay, etc., which results in packet loss and retransmission. However, waiting for the expiration of the Interest lifetime in order to resend the Interest is not appropriate for delay-sensitive applications that require high throughput and minimum delay. In order to provide delay-sensitive applications with better Quality of Service (QoS), early network congestion prevention is very important in CCN.

For the congestion control mode defined in [7], in receiver-driven proposals [8], [9], congestion detection and traffic shaping are carried out only at consumer nodes/receivers, which have disadvantages of not considering multi-source feature of CCN. On the other hand, hop-by-hop proposals [10]–[12] have the advantage of detecting congestion and adjusting traffic rate more effectively than the receiver-driven proposals, where congestion control and traffic shaping are carried out at each intermediate node. However, hop-by-hop control mechanism without receiver-driven control mechanism is not sufficient to control congestion and traffic rate accurately. To overcome this challenge, hybrid congestion control approaches [13]–[15] combine both receiver-based and hop-by-hop approaches that can be used to adjust the traffic rate more accurately than separated receiver-driven and hop-by-hop proposals. Nevertheless, in hybrid congestion control, the coordination of receiver-driven congestion control at consumer nodes and hop-by-hop congestion control at intermediate nodes have not been addressed so far in CCN [7].

B. HOW CONGESTION OCCURS IN CCN?

In CCN, congestion may occur in the network as packets overflow the transmission buffer associated to outgoing face/interface [16]. In other words, packets arrive faster than the router's processing capacity (or more than the link capacity), which causes buffer fullness, packet loss, and Interest retransmission.

In both TCP/IP network and CCN, the buffer is used to accommodate the transient packets in order to ensure that the network links do not become idle [17]. As the buffer introduces queuing delay, buffer monitoring is needed to guarantee that the router stays within a region of low delay and high throughput, so that it does not overflow, which results in packet drop.

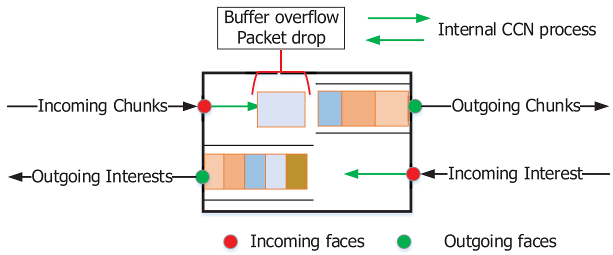

FIGURE 1. CCN buffer.

However, as described in Fig. 1, major cause of CCN buffer overflow or congestion is the incoming Chunks from Content Provider (CP) or upstream routers rather than Interest packets, which are the results of outgoing Interest packets generated by consumers [10]. Therefore, to prevent buffer overflow, we focus on regulating Interest forwarding. As the Chunk returns in the reverse path of Interest packet, regulating Interest traffic contributes in regulating returning Chunks traffic.

For simplifying our illustration in the Fig. 1, we omit CS, PIT and FIB tables, which are involved in the internal CCN router process. Inside the router, after missing Chunks in CS and matching PIT entries, the incoming Interest packets are forwarded to the outgoing faces through the use of FIB table. On the other hand, incoming Chunks are forwarded to the outgoing faces, when there are corresponding entries in PIT; thus the Chunks have to be forwarded in the reverse paths of Interest packets. Therefore, CS and PIT reduce the network traffic by locally saving most frequently requested chunks and already forwarded Interest packets.

C. CHALLENGES IN CCN CONGESTION CONTROL

Congestion control in TCP/IP network is totally different from CCN congestion control. Some key challenges that prevent existing congestion control mechanisms mainly proposed for Transmission Control Protocol (TCP) [18], [19] for being deployed in CCN are summarized below:

- In CCN, each content is partitioned into Chunks, where one Interest packet is emitted to request one Chunk. For the Chunks returning to the consumers, any node in the transmission path can cache them for serving similar requests in the near future from other consumers. In other words, Chunks of the same content can be retrieved from different nodes and paths with different Round Trip Times (RTTs) [20]. This RTT variation prevents existing traditional measurements of RTT in TCP from being utilized in CCN.
- In TCP, congestion control is based on end-to-end communication between predefined source and

the destination. On the other hand, in CCN, due to in-network caching along the transmission path, Chunk keeps changing its location [21], and this results in change of source of the content.

- In TCP, during congestion period, for fast retransmission of the lost packet, when a node receives a third duplicated acknowledgment, it considers that the packet was lost and retransmits the packet. On the other hand, in CCN, there is no such duplicated acknowledgment [7]. When the consumer node does not receive the corresponding Chunk of submitted Interest within the timeout period or Interest lifetime, it considers that the Interest packet was lost, the consumer node retransmits the Interest packet. The returning Chunk acts as the acknowledgment of the received Interest packet [22].

D. CONTRIBUTIONS

To overcome the above challenges faced by existing end-to-end based congestion control mechanisms, we propose a Novel Cooperative and Fully-distributed Congestion Control Mechanism for Content Centric Networking (NCFCC). NCFCC is capable to prevent congestion before it occurs by monitoring buffer utilization. NCFCC combines consumer-driven and hop-by-hop interest shaping into one hybrid congestion control solution.

Our key contributions can be summarized as follow:

- To prevent congestion, without waiting for the Interest lifetime to expire or timeout, we propose Cooperative and Memory-efficient Token Bucket (CMTB) algorithm, which is based on local measurements that are applied to each intermediate node (hop-by-hop). CMTB algorithm monitors buffer size and controls the rate at which packets are injected into the network, in order to ensure that the CCN router buffer does not overflow (i.e., when it starts dropping the packets), as well as to avoid underflow (i.e., when it makes the link idle or underutilized).
- To complement CMTB algorithm, on consumer node, we proposed Fully-Distributed Congestion Control (FDCC) algorithm as receiver-based congestion control. FDCC algorithm is applied to each consumer node. We consider the consumer side as the actual position from where the congestion can be controlled in the most effective way. Upon receiving congestion information, consumer node reduces the sending rate through the use of FDCC. The FDCC guarantees that the consumer node does not send more packets than the network capacity.
- We have carefully evaluated our proposal through simulation and by comparing it with similar proposals. The simulation results reveal that our proposed model is capable of significant performance improvements, with higher throughput than other existing proposals in the literature.

The remainder of the paper is organized as follows: In Section II, we discuss some related works, while Section III presents our system model. Section IV discusses in detail proposed congestion control algorithms, while Section V provides evaluation of the proposed congestion control mechanism. We conclude the paper with a review of the implications and future research directions in Section VI.

II. RELATED WORK

In this section, we discuss some important works that are related to our work. We classify them into three categories: (i) congestion control mechanisms in CCN, (ii) relationship between buffer and congestion control, and (iii) traffic shaping.

(i) Congestion control mechanisms in CCN: Park *et al.* [11] highlighted that the major cause of the congestion in CCN is the Data packets rather than the Interest packets. They therefore propose a congestion control algorithm that considers content caching, upstream link bandwidth, Data packets received from upstream link, and the Interest packets received from downstream link in order to prevent congestion by adjusting outgoing Interest rate that needs to be forwarded to the upstream link.

Mahdian *et al.* [12] proposed a rate-based multipath-aware congestion control scheme for ICN networks called MIRCC. In MIRCC, based on feedback information in Data messages, each node calculates per-link rates. MIRCC allows the utilization of multipath, where network flow utilizes the network resources along all the available paths to it. Furthermore, the feedback-based congestion control is also proposed in [8], where upon the reception of each Interest, the router verifies that it has the next Chunk(s) that is intended to be requested in the future. If this is confirmed, then it appends the information about it to the returning Chunk. Based on this process, the consumer node maintains multiple timeout values for each path in order to predict the location of the Chunks before being requested, and accurately estimates the retransmission timeouts.

Carofiglio *et al.* [13] reported that when the Interest is sent, the receiver sets a timer. If the timer expires without receiving the Data packet, it is assumed that congestion has occurred, and decreases the receiver window. The authors therefore propose a model, where at every outgoing face, the Interest control rate is carried out with the help of a credit counter. When the flow is not bottlenecked, the Interest packet is forwarded to the upstream node; otherwise, through the use First-In First-Out (FIFO) queuing model, the Interest is queued in FIFO with drop tail. The same authors (Carofiglio *et al.*), in [14], formulated a global optimization problem, which aims at maximizing throughput and minimizing network cost. The authors decomposed the global optimization problem into two subproblems for congestion control and request forwarding.

Other alternatives have also been proposed. For instance, Oueslati *et al.* [15] developed a model based on per-flow queuing and overload control. With the help of Deficit Round Robin (DRR), the router drops packets from the longest queue in the event of buffer overflow. When a packet is dropped, the packet payload is discarded and its header is modified and returned to the consumer as a signal of congestion. This helps the consumer node to detect the packet loss and retransmit it without waiting for the timeout. Furthermore, for reliable Interest retransmission in multicast environment, Stais *et al.* [9] proposed a retransmission-based control protocol for ICN, where multicast tree is used for sending the feedback to the consumer.

Given the continued complexity of this problem, still other solutions have been developed. In the model proposed by Rozhnova and Fdida [10], each node monitors the level of Chunks in the router's transmission buffer, computes the associated Interests, and maintains the transmission queue around the fixed threshold r. When the number of packets in the queue is lower than the buffer threshold r, the node increases the shaping rate. Otherwise, the node decreases shaping rate with the help of a shaping delay in which the different Interest packets have to meet. In the case of buffer overflow, the Interest shaping allows for an unanticipated drop of the packets.

(ii) *The relationship between buffer and congestion control*: Wischik and McKeown [23] analyzed in detail the router buffer size, and how large it needs to be. It was revealed that larger buffers increase RTT at a congested link because of a long queuing delay. On the other hand, in small buffers, burst flows cause buffer fullness, which results in packet drop. The rule of thumb here comes as a solution for keeping the congested link busy and having maximum throughput, but no matter how large a buffer is, TCP always causes a buffer overflow and packet dropping as a signal of congestion.

(iii) *Traffic shaping*: Kidambi *et al.* [24] proposed an algorithm called the Dynamic Token Bucket (DTB) for bandwidth management, which uses token bucket policing for fair bandwidth allocation to competing flows. The DTB is a modified version of the token bucket algorithm, in which each flow is allocated a specific token bucket. The authors used a fair rate to define the capacity of the token bucket, where the capacity of the token bucket at time t is equal to the fair rate, which depends on link capacity divided by the number of active flows.

Specifically, the novelties of our proposal over related works are: *(i)* Rather than taking measurement when congestion is happening, NCFCC prevents congestion before it happens through monitoring buffer utilization, and adjusting traffic rate. *(ii)* NCFCC combines inseparable hop-by-hop (CMTB) and receiver-based (FDCC) congestion control algorithms into one hybrid congestion control mechanism, where FDCC (at consumer node) prevents congestion and adjusts the traffic rate based on reduce sending rate (RSR) messages generated by CMTB (at Intermediate node). *(iii)* The coordination among CMTB algorithms implemented in different intermediate nodes is based on RSR message exchange. Furthermore, RSR message coordinates the communication between FDCC implemented at consumer node with its upstream node at which CMTB is implemented, *(iv)* RSR helps consumer nodes to adjust Interest sending rate without waiting for the expiration of an Interest's lifetime and timeout. RSR also helps the intermediate nodes to adjust per-link traffic rate based on queue length.

III. SYSTEM MODEL

In this section, we describe in detail our system model established as the foundation for the proposed Congestion Control mechanism in CCN.

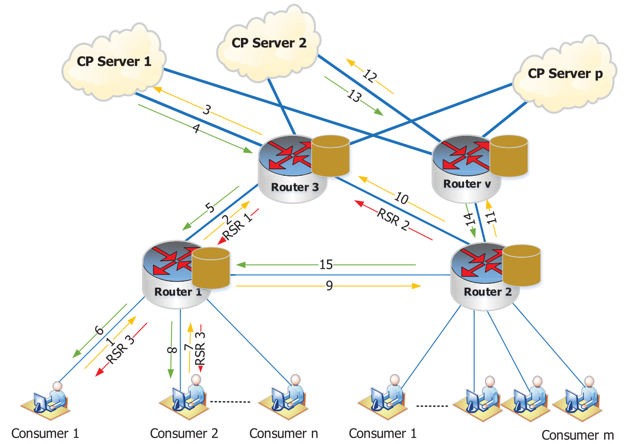

FIGURE 2. System model.

Content Provider (CP) Server: In our system model depicted in Fig. 2, we consider $\mathcal{P} = \{1, \dots, P\}$ as a set of CP servers that are responsible for the actual distribution of contents from the content providers.

Consumers and Interest packets generation: Let us consider $\mathcal{M} = \{1, \dots, M\}$ as a set of all consumers, where consumer node $m \in \mathcal{M}$ is associated with cache of capacity C_m for caching the content. Each consumer node $m \in \mathcal{M}$ uses FDCC algorithm for generating and adjusting Interest packets. The consumer node starts generating, and sending Interest packets in the network with initial window size w_m. For the chunk i belongs to the content catalog Γ requested for the first time, the consumer node gets i from CP server, and i gets copied in the caches associated with the nodes available in the transmission path as it returns to the consumer (step 1 to 6). Similar requests from other consumers can be served from caches available in the transmission path with minimized delay rather than retrieving the content from the CP server (step 7 to 8).

Router: We denote $\mathcal{V} = \{1, \dots, V\}$ as a set of all routers available in the transmission path between consumer and CP server, where each router $v \in \mathcal{V}$ is associated with cache of capacity C_v for caching the content. Moreover, each router is connected to some other node(s) via intermediate link(s). We denote $\mathcal{J}_v = \{1_v, \dots, J_v\}$ as a set of links associated to node v, where each link has capacity C_j, for $j \in \mathcal{J}_v$. Furthermore, we denote $\mathcal{K}_v = \{1_v, \dots, K_v\}$ as the set of the faces associated to router v, connecting v to other neighboring node(s) through the use of intermediate link(s) $j \in \mathcal{J}_v$.

TABLE 1. Key notations.

Symbol	Definition
$\mathcal{P}$	Set of CP servers, $\lvert\mathcal{P}\rvert = P$
$\mathcal{V}$	Set of routers, $\lvert\mathcal{V}\rvert = V$
$\mathcal{M}$	Set of consumers, $\lvert\mathcal{M}\rvert = M$
z_m	Traffic rate generated by consumer node $m \in \mathcal{M}$
U_m	Utility function of the consumer m
w_m	Initial window size for consumer m
$\mathcal{J}_v$	Set of all links associated to node $v \in \mathcal{V}$, $\lvert\mathcal{J}_v\rvert = J_v$
$\mathcal{M}(j)$	Set all consumers using the link j, $\mathcal{M}(j) \subset \mathcal{M}$
$\gamma_j(z)$	Traffic load on link j
$\mathcal{K}_v$	Set of all faces associated to node v, $\lvert\mathcal{K}_v\rvert = K_v$
$N_v(t)$	Number of flows at time t in node v
B_v	Total buffer size at each node v
B_k	Buffer size associated to the face $k \in \mathcal{K}_v$
T_k	Buffer threshold at each face k
T_v	Minimum buffer threshold at each router v
C_j	Link capacity, for $j \in \mathcal{J}_v$
C_v	Cache capacity associated to router v
$R_k(t)$	Fair rate for outgoing packets at face k
$E[\alpha_v]$	Expected number of packets in node $v \in \mathcal{V}$
$E_k[L_k]$	Expected number of packets in queue associated to the face k
p_{B_k}	Loss probability,
ρ_v	Traffic intensity at router v
Γ	Content catalog
q	Scaling parameter
i	Data object/chunk $i \in \Gamma$
l_i	Length of the packet $i \in \Gamma$
ω_j	Link price, for $j \in \mathcal{J}_v$
RTT	Round Trip Time
VRTT	Virtual Round Trip Time
RTO	Retransmission Time-Out

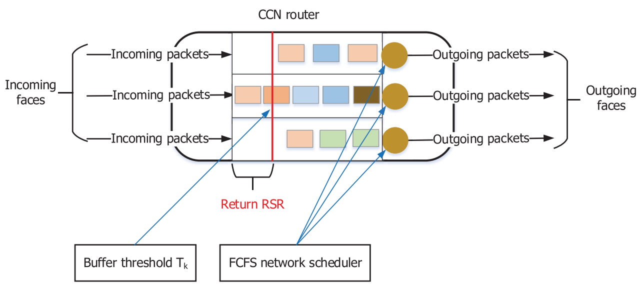

FIGURE 3. Router buffer model.

Router Buffer: We consider that each router v has a buffer/memory B_v shared among its K_v face(s). As described in Fig. 3, each router needs to allocate buffer size fairly to each outgoing face. We denote B_k as the buffer associate to the outgoing face $k \in \mathcal{K}_v$, and T_k as the buffer threshold. The incoming packets are assigned to the buffer and served in the order they arrive through the use of First Come First Serve (FCFS) network scheduler, where the packet scheduler arranges the transmission sequence of incoming packets. When all buffers associated to K_v face(s) reach the threshold, the node sends congestion alert message (RSR) to its downstream node. On received congestion alert message, the downstream node reduces the traffic rate on affected outgoing face through the use CMTB, by allocating new incoming Interest packets to other outgoing face (step 9 to 15 in Fig. 2). On the other hand, consumer node reduces the traffic rate through the use of FDCC.

CCN flow: We use "flow" to denote the Interest and corresponding chunk pairs, where flows are distinguished by looking the prefix names that are common to all Interests/chunks of the same object (video stream, music file, stored document, etc.) as described by Oueslati *et al.* in [15]. The incoming packets are classified into flows, and queued to the outgoing buffer based on their prefix names. Furthermore, to classify the incoming packets into different flows, every incoming Interests/chunks with similar prefix name are considered to belong to the same flow. Using prefix names to classify packets into flows does not create any problem in CCN, as Interest packet and its corresponding Data packet always use the same path(s). However, based on buffer utilization, Interests/chunks of the same flow may be queued to the different outgoing faces.

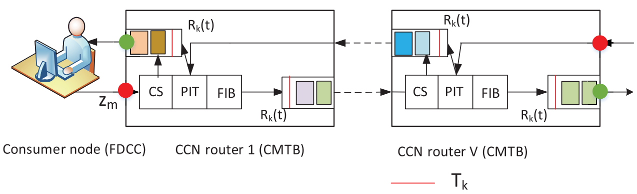

FIGURE 4. Relationship between CMTB and FDCC.

IV. NOVEL COOPERATIVE AND FULLY-DISTRIBUTED CONGESTION CONTROL MECHANISM

In this section, we discuss in details our NCFCC, which combines inseparable hop-by-hop (CMTB) and receiver-based (FDCC) congestion control algorithms into one hybrid congestion control mechanism. As described in Fig. 4, CMTB controls the rate ($R_k(t)$) at which packets are injected into the network in the intermediate nodes, while FDCC controls the traffic (z_m) rate in consumer nodes based on Chunks and congestion information received. We conclude the section with communication between CMTB and FDCC, through the use of RSR, which serves as the link between CMTB and FDCC algorithms for exchanging congestion information between neighboring nodes.

A. COOPERATIVE AND MEMORY-EFFICIENT TOKEN BUCKET (CMTB)

In this subsection, we present a new algorithm CMTB for CCN, which prevails over the simplicity and efficiency of the DTB in terms of computation and fair bandwidth allocation. CMTB deals with fair buffer resource allocation, and traffic shaping.

1) CMTB MODELING

In CMTB, we use an $M/M/1/B$ finite buffer queuing system [25], where the packet arrives at router $v \in \mathcal{V}$ according to the Poisson process with arrival rate λ_v. The departure process also follows the Poisson process with service rate μ_v.

Let us consider B_v as the total buffer of each CCN router $v \in \mathcal{V}$, where buffer space B_v is shared among K_v number of faces/interfaces associated to router $v \in \mathcal{V}$. For fair sharing of buffer space B_v, each face $k \in \mathcal{K}_v$ can utilize B_k as maximum buffer space, where B_k is given by:

$$B_k = \frac{B_v}{K_v}. \tag{1}$$

In $M/M/1/B$, we introduced T_k as buffer threshold associated to the face $k \in \mathcal{K}_v$, and T_v as common minimum buffer threshold used in all K_v number of faces associated to the node $v \in \mathcal{V}$. The network administrator assigns T_v to each node. From T_v, each node computes buffer threshold T_k dynamically as follows:

$$T_k = \max\{(B_k - E[L_k]), T_v\}, \tag{2}$$

where $E[L_k]$ is the expected number of packets in queue or queue occupancy associated to the face $k \in \mathcal{K}_v$.

In a finite buffer system, a new incoming packet gets dropped when the buffer overflows. In this paper, we use per-face buffer, and a node can only support the traffic at face $k \in \mathcal{K}_v$ that is equal to buffer size B_k associated to the face k, which is limited. To prevent packet drop, in $M/M/1/B$, when the buffer threshold T_k is reached, the node requests its downstream node(s) to reduce the traffic rate.

To model the buffer utilization, let us denote $\rho_v = \frac{\lambda_v}{K_v \mu_v}$ as the traffic intensity or utilization in node v of K_v face(s), where the probability of α_v packets in node $v \in \mathcal{V}$ is equal to $p_{\alpha_v} = \frac{(1-\rho_v)(\rho_v)^{\alpha_v}}{1-\rho_v^{B_k+1}} = \rho_v^{\alpha_v} p_0$. We consider p_0, the probability that the new incoming packet finds the buffer B_k empty, where $p_0 = \frac{1-\rho_v}{1-\rho_v^{B_k+1}}$. However, the probability of an incoming packet sees B_k packets (buffer is full) is $p_{B_k} = \frac{(1-\rho_v)(\rho_v)^{B_k}}{1-\rho_v^{B_k+1}} = \rho_v^{B_k} p_0$.

For a larger buffer size, each node maintains a loss probability (p_{B_k}) of less than 1% [25]. From p_{B_k}, the router computes the required buffer for each incoming packet. Optimal buffer size selection is very important, because a smaller buffer has low queuing delay but overflows quickly, and packets get dropped easily. On the other hand, a large buffer overcomes the packet loss problem, but it affects the delay-sensitive application because of the high queuing delay. To precisely define the appropriate buffer size, there are many parameters that need to be considered, such as link capacity, consumer window size, and RTT, but consideration of these parameters needs to be well balanced.

In our previous work [22], we opted for the optimal face buffer B_k needed for the L-model, as was also proposed by Prasad *et al.* [26]. Even if the optimal buffer associated with face $k \in \mathcal{K}_v$ for the L-model is large, and allow for the router to prevent the packet drop, it increases the delay experienced by the packets. In this paper, we opted Bandwidth Delay Product (BDP) presented in equation (3), where the buffer size is considered to be relatively small, which reduces delay.

Each node $v \in \mathcal{V}$ monitors buffer utilization (by using CMTB algorithm that will be introduced later in this subsection), and provides a fair share rate to transient packets in order to ensure that the buffer does not become full and drops the packets, and so that there is not an underflow leading to an idle or underutilized link.

The fair share rate depends on the number of flows, link capacity, buffer size, queue occupancy, and RTT. We define RTT_i as the length of time required for sending Interest packet, and receiving the corresponding Chunk $i \in \Gamma$. However, in CCN, nodes on transmission path can cache the Chunks, which results in the change of Chunk's locations and RTTs. Thus, traditional measurements based on RTT in the TCP/IP network cannot work properly in CCN, we therefore embraced the idea of Virtual Round Trip Time (VRTT) [21]. $VRTT_v$ is defined as an average time for sending Interest packets and receiving correspondent Chunks at node v.

From $VRTT_v$, and based on bandwidth delay product rule [17], we formulate the buffer needed to the face $k \in \mathcal{K}_v$ as follows:

$$B_k = C_j \times VRTT_v. \tag{3}$$

We consider that the needed buffer size B_k in (3) has to be less or equal to the buffer space computed in (1), i.e., (1) defines buffer upper bound.

In our model, ρ_v determines the relationship between p_{B_k} and B_k at router $v \in \mathcal{V}$. For heavy network traffic, ρ_v tends to approach 1, and during that period, the network has better throughput. On the other hand, when ρ_v approaches 0, the link is becoming empty. In order to maintain a stable network, our CMTB maintains $0 \leq \rho_v \leq 1$.

Based on the above defined B_k (3), each node assigns a fair rate $R_k(t)$ to all outgoing packets passing through its face. For computing $R_k(t)$, let $E[\alpha_v] = \frac{\rho_v}{1-\rho_v} - \frac{B_k+1}{(1-\rho_v^{B_k+1})}\rho_v^{B_k+1}$ be the expected number of packets in node $v \in \mathcal{V}$, and $E_k[L_k] = E[\alpha_v] - \frac{\lambda_v}{\mu_v}$ be the expected number of packets in queue or queue occupancy for each face $k \in \mathcal{K}_v$. The fair rate $R_k(t)$ for each outgoing face becomes:

$$R_k(t) = \frac{C_j(t)}{N_v(t)} + \frac{B_k - E_k[L_k]}{VRTT_v N_v(t)} \tag{4}$$

The fair rate question 4 is composed of two parts: the left hand side $\frac{C_j(t)}{N_v(t)}$ represents the traffic rate at time t that the node can inject into the network without accommodating the transient packets in buffer, while the right hand side $\frac{B_k - E_k[L_k]}{VRTT_v N_v(t)}$ represents the actual buffer size that is used to store transient incoming packets per each $VRTT_v$, while waiting the link to become available for sending the packets.

2) CMTB ALGORITHM FOR INTERMEDIATE NODES

In our CMTB algorithm, presented in Algorithm 1, given the face $k \in \mathcal{K}_v$ and the incoming packet (Interest or Chunk) as an input (at line 1), node checks the prefix name, if the Interest $i \in \Gamma$ is not congestion alert packet (at line 3), node lookups in CS and PIT. In case of miss in both tables, node $v \in \mathcal{V}$ calculates the fair rate $R_k(t)$ for outgoing Interest packets (at line 6). If the length of the packet i (l_i) is less or equals to

Algorithm 1 Cooperative and Memory-Efficient Token Bucket

1: **Preconditions:** A node receives Interest i or Chunk i at inface $k \in \mathcal{K}_v$;
2: At Interest i reception (prefix, inface $k \in \mathcal{K}_v$);
3: **if** Interest i's prefix $\neq$ /RSR **then**
4:   **if** CS_miss && PIT_miss **then**
5:     FIB_lookup(prefix);
6:     Calculate $R_k(t)$ using equation (4);
7:     **if** $(l_i \leq R_k(t))$ **then**
8:       **if** $E_k[L_k] < T_k$ **then**
9:         Allocate Interest i to buffer B_k, forward_interest (outface $k \in \mathcal{K}_v$);
10:       **else** {Hint: Increment k (i.e., check another outface)}
11:         FIB_lookup(prefix);
12:         Allocate Interest i to buffer B_k, forward_interest (outface $k \in \mathcal{K}_v$);
13:         Repeat *steps* 7 *to* 12, till all K_v outfaces are checked;
14:         **if** $E_k[L_k] \geq T_k$ and $E_k[L_k] \leq B_k$ in all K_v outfaces **then**
15:           Allocate Interest i to buffer B_k, forward_interest (outface $k \in \mathcal{K}_v$;
16:           Generate new Interest i with prefix /RSR;
17:           Return Interest i (/RSR, inface $k \in \mathcal{K}_v$);
18:         **end if**
19:       **end if**
20:     **else** {Hint: $l_i > R_k(t)$}
21:       Drop Interest i;
22:     **end if**
23:   **end if**
24: **else** {Hint: Interest i's prefix = /RSR}
25:   Increment k;
26:   Drop Interest i;
27:   Repeat *steps* 11 *to* 17
28: **end if**
29: At Chunk i reception (prefix, inface $k \in \mathcal{K}_v$);
30: **if** PIT_miss **then**
31:   Drop Chunk i;
32: **else** {Hint: PIT_hit}
33:   Calculate $R_k(t)$ using equation (4);
34:   **if** $(l_i \leq R_k(t))$ **then**
35:     Allocate Chunk i to buffer B_k,
36:     Forward_chunk (outface $k \in \mathcal{K}_v$);
37:   **else** {Hint: $l_i > R_k(t)$}
38:     Drop Chunk i;
39:   **end if**
40: **end if**

$R_k(t)$ (at line 7), node $v \in \mathcal{V}$ checks whether the buffer has reached the threshold (at line 8).

At lines $9-19$, Interest packet requesting Chunk $i \in \Gamma$ is placed in the buffer B_k associated with the face $k \in \mathcal{K}_v$ until B_k reaches the threshold. When B_k reaches the threshold T_k, the node $v \in \mathcal{V}$ checks other buffers (in case the node has many faces), and allocates the new incoming Interest packets to another outgoing face's buffer, based on its state of fullness. When all buffers reach the threshold, the node returns RSR message to its downstream node (at line 17). Furthermore, at lines $20-22$, when $l_i > R_k(t)$, the node drops Interest packet.

At lines $24-28$, upon reception of Interest packet with prefix name /RSR, the node increments k and allocates the new incoming Interest packets to another outgoing face's buffer.

At lines $29-36$, for each Chunk $i \in \Gamma$, without violating the original CCN operational principle, where the router returns the Chunk $i \in \Gamma$ to the consumer in the reverse path of the Interest packet, each node $v \in \mathcal{V}$ allocates Chunk i to the buffer B_k associated to the face $k \in \mathcal{K}_v$ that was used to request the Chunk i through the use of PIT. Our algorithm does not allocate returning Chunks to other face(s) rather than the face used to request Chunk i, i.e. in the absence of corresponding PIT entry, it drops Chunk packet i.

At lines $37-39$, when buffer used to request Chunk i is full $(l_i > R_k(t))$, the node drops the packet, which is very rare due to the buffer threshold utilization, where on reception of congestion alert message (RSR), the downstream node allocates the new incoming Interest packets to another outgoing faces. However, when the downstream node fails to reduce traffic rate, the same procedure continues until the consumer node reduces sending rate through the use of FDCC discussed below.

B. FULLY-DISTRIBUTED CONGESTION CONTROL (FDCC)

The main objective of the congestion control algorithm is to detect congestion at early stage, or to prevent it before it happens. The intermediate nodes prevent congestion through the use of CMTB described in Section IV-A. However, when all of the intermediate nodes in the network fail to prevent the congestion by using CMTB, the consumer node needs to take action in preventing congestion. The consumer node attempts to prevent congestion through the use of FDCC, which reduces the traffic rate by making sure that the consumer node does not send more packets than the network capacity.

In this subsection, we present in details FDCC algorithm for Interest lifetime estimation, and dynamic Interest traffic regulation based on received congestion alert message, which is implementable in the consumer node.

In CCN, there is no duplicated acknowledgment for packets not received. The returning Chunk acts as the acknowledgment of the received Interest packet, based on which the consumer node $m \in \mathcal{M}$ keeps updating the $VRTT_m$, and Retransmission Time-Out (RTO). When the consumer node does not receive the corresponding Chunk of submitted Interest within the RTO_i, it considers that the Interest packet was lost, the consumer node retransmits the Interest packet after RTO_i or expiration of Interest lifetime. We consider RTO_i as the length of time that a node needs to wait until it concludes that the submitted Interest has failed to return a corresponding

Chunk $i \in \Gamma$, or the absence of the requested Chunk. To retransmit unsatisfied Interest packet, the consumer node has to wait the expiration of three $VRTT_m$.

In most circumstances, users request chunks (or a large portion) of the content continuously (especially in the case of audio/video). We assume that the users may find the consecutive Chunks of the content in the nearby nodes in transmission path, and it is quite possible that the users get consecutive Chunks of the content from the same node [27]. This has motivated us to borrow ideas from TCP AIMD [28] and retransmission [29], and modify them for CCN.

1) FDCC MODELING

To model our FDCC, let us denote $z = (z_m)$ as a vector of traffic generated consumer nodes, and $\gamma_j(z)$ as the traffic load on link $j \in \mathcal{J}_v$ connecting consumer nodes to router $v \in \mathcal{V}$. The traffic load on link $j \in \mathcal{J}_v$ is the sum of all traffic generated by consumer nodes using that link, where $\mathcal{M}(j)$ is the set of all consumers using the link j, for $\mathcal{M}(j) \subset \mathcal{M}$, such that

$$\gamma_j(z) = \sum\nolimits_{m \in \mathcal{M}(j)} z_m(1 - h_i) \quad (5)$$

where h_i is the cache hit probability for the Chunk $i \in \Gamma$ requested by consumer node m. For the simplicity of our model, we assume that the node sends traffic to the link $j \in \mathcal{J}_v$ when it misses Chunk in its CS. The total traffic passing through the link $j \in \mathcal{J}_v$ must be less than or equal to the link capacity C_j.

$$\sum\nolimits_{m \in \mathcal{M}(j)} \gamma_j(z_m) \leq C_j. \quad (6)$$

To investigate a fair resource/link capacity allocation, let us define a utility function $U_m(z_m)$ as a modeling tool for analyzing our FDCC, where utility function indicates how happy the consumer $m \in \mathcal{M}(j)$ is, based on the resource/traffic rate z_m assigned to him. The resource allocation becomes fair, when it maximizes the sum of the utilities of all consumers, subjects to the link capacity constraint, which is referred as social welfare maximization.

For Network Utility Maximization (NUM) [30], [31], the optimization problem becomes:

$$\begin{aligned} &\text{maximize} \sum\nolimits_{m \in \mathcal{M}(j)} U_m(z_m) \\ &\text{subject to} \sum\nolimits_{m \in \mathcal{M}(j)} \gamma_j(z_m) \leq C_j \\ &\qquad z_m \geq 0, \quad \forall m \in \mathcal{M}(j). \end{aligned} \quad (7)$$

The above formulated problem is convex optimization problem, which can be decomposed into subproblems through the use of dual decomposition technique. To solve (7), the Lagrangian can be written as:

$$L(z, \omega) = \sum\nolimits_{m \in \mathcal{M}(j)} U_m(z_m) + \sum\nolimits_j \omega_j(C_j - \gamma_j(z)), \quad (8)$$

where ω_j is the positive weights, namely Lagrange multiplier, which is used as the link price, for $j \in \mathcal{J}_v$. The sum of all prices along the links used by consumer m is denote $\tau_m(\omega)$, where $\omega = (\omega_j)$ is the vector of the link prices, such that

$$\tau_m(\omega) = \sum\nolimits_{j:m \in \mathcal{M}(j)} \omega_j. \quad (9)$$

The maximization of Lagrangian problem L over z_m is independently computed by each consumer node m, and is dependent on Lagrangian multiplier ω_j used. The Lagrangian problem becomes:

$$L(z, \omega) = \sum\nolimits_{m \in \mathcal{M}(j)} U_m(z_m) - \tau_m(\omega)\gamma_j(z) + \sum\nolimits_j \omega_j C_j. \quad (10)$$

The Lagrangian dual function can be written as

$$g(\omega) = \max_z L(z, \omega), \quad (11)$$

where the solution to (11) can be represented as follows:

$$z_m^*(\omega) = \text{argmax } (U_m(z_m) - \tau_m(\omega_j)\gamma_j(z)). \quad (12)$$

Minimizing Lagrangian dual function g over ω is done by using gradient method with β as step size, where $C_j - \gamma_j(z)$ is the gradient of each ω_j. Furthermore, for simplifying our notation, we denote $z'_m = z_m(1 - h_i))$, then we have:

$$\omega(t) = (\omega(t-1) - \beta(C_j - \sum\nolimits_{j:m \in \mathcal{M}(j)} z'^*_m(\omega_j))). \quad (13)$$

Coming back to our utility function, the optimum consumer' traffic rate needs to satisfy $\frac{\partial U_m}{\partial z'_m} = 0, \forall m \in \mathcal{M}(j)$. In other words, $U'(z'_m) - \tau_m(\omega) = 0, \forall m \in \mathcal{M}(j)$. Since $U'(z'_m)$ is invertible, z'^*_m can be written as:

$$z'^*_m(t) = U'_m(z'_m)^{-1} - \tau_m(\omega, t), \quad \forall m \in \mathcal{M}(j). \quad (14)$$

Let us consider q as scaling parameter, and for each Chunk received, the consumer node sets scaling parameter equals to 1. When consumer node receives congestion alert from the network (the congestion alert message is discussed in Section IV-C), the consumer node sets scaling parameter equals to 1/2. On timeout or on expiration of the Interest lifetime, the consumer node sets scaling parameter equals to 1/3, and retransmits unsatisfied Interest(s). Each consumer decides on transmission rate as follows:

$$z'^*_m(t) = q(U'_m(z'_m)^{-1} - \tau_m(\omega, t)), \quad \forall m \in \mathcal{M}(j). \quad (15)$$

2) FDCC ALGORITHM FOR CONSUMER NODES

In FDCC, the consumer node $m \in \mathcal{M}(j)$ starts sensing the network with the initial window size w_m, i.e, generating Interest packets with initial window w_m, where the latter defines the initial transmission rate z_m. Initial Interest lifetime is set to 4 seconds, which is the default value for CCN [32], and initial RTO_i is set also to be 4 seconds (at line 2). At lines 3 − 9, upon the reception of each requested Chunk, the consumer node calculates $VRTT_m$, updates Interest lifetime and RTO_i. In additional, consumer node keeps increasing the window size by $1/w_m$ above its current window size w_m and schedules to transfer next Interest. In other words, in each Chunk received consumer node increases 1 Interest Packet after each $VRTT_m$. At the lines $12-15$, when consumer node receives congestion alert from the network, the consumer node reduces one half of the current w_m and schedules to transfer next Interest packet. From lines $17-21$, on Interest lifetime expiration or timeout, the consumer node $v \in \mathcal{V}$ doubles RTO_i and Interest lifetime, and retransmits unsatisfied Interest packet.

Algorithm 2 Fully-Distributed Congestion Control

1: **Preconditions:** A node generates Interest i, receives Interest i or chunk i at inface $k \in \mathcal{K}_v$;
2: The node generates Interest packets with initial window w_m and forwards them to outface $k \in \mathcal{K}_v$; $Interest\_lifetime \leftarrow 4s$, $RTO_i \leftarrow 4s$;
3: At chunk i reception (prefix, inface $k \in \mathcal{K}_v$);
4: **if** PIT_miss **then**
5:     Drop Chunk i;
6: **else** {Hint: PIT_hit}
7:     compute $VRTT_m$, $Interest\_lifetime \leftarrow 3VRTT_m$, $RTO_i \leftarrow 3VRTT_m$;
8:     $w_m = w_m + 1/w_m$;
9:     *Schedule_Next_Interest_Transfer*;
10: **end if**
11: At Interest i reception (prefix, inface $k \in \mathcal{K}_v$);
12: **if** Interest i's prefix = /RSR **then**
13:     Drop Interest i;
14:     $w_m = w_m/2$;
15:     *Schedule_Next_Interest_Transfer*;
16: **end if**
17: **if** RTO_i expires **then**
18:     $w_m = w_m/3$;
19:     Double *Interest_lifetime* and RTO_i;
20:     *Retransmit_Unsatisfied_Interest*;
21: **end if**

For more analysis of our Algorithm 2, on congestion alert message reception, the change in window size [30] can be written as:

$$z'_m(t)(1-\tau_m(\omega,t))\frac{1}{w_m(t)} - z'_m(t)\tau_m(\omega,t)\frac{w_m(t)}{2}. \quad (16)$$

After reducing current window size w_m, consumer node starts again to increase the window size by 1 Interest packet per $VRTT_m$ for each chunk received. Therefore, for simplifying our notation, we denote $d_m = VRTT_m$, and the equation (16) becomes:

$$z'_m(t+1) = z'_m(t) + \frac{1-\tau_m(\omega,t)}{{d_m}^2} - \frac{1}{2}z'^2_m(t)\tau_m(\omega,t). \quad (17)$$

On timeout or on expiration of the Interest lifetime, the consumer node reduces one third of the window size w_m, and retransmits the unsatisfied Interest. Then, the equations (16) and (17) become:

$$z'_m(t)(1-\tau_m(\omega,t))\frac{1}{w_m(t)} - z'_m(t)\tau_m(\omega,t)\frac{w_m(t)}{3}, \quad (18)$$

$$z'_m(t+1) = z'_m(t) + \frac{1-\tau_m(\omega,t)}{{d_m}^2} - \frac{1}{3}z'^2_m(t)\tau_m(\omega,t). \quad (19)$$

3) COMPARISON BETWEEN TCP CONGESTION CONTROL AND FDCC

Since the CCN network is different from TCP/IP, the traditional TCP congestion control in conventional networks cannot be directly applied to CCN. Thus, in summary, there are some key differences between TCP and our proposed FDCC as follows:

- TCP infers congestion based on packet loss or RTO, while FDCC infers congestion based on received congestion alert message generated by CMTB, which is the result of the traffic load in upstream nodes. Congestion control based on RTO is not reliable in CCN, especially for delay-sensitive applications.
- TCP is based on established communication between sender and receiver (end-to-end communication). In case of packet loss, the receiver sends duplicate acknowledgments to the sender informing the expected packet to receive, so that the sender can retransmit the packet. In contrast, in FDCC, there is no such duplicated acknowledgments. When the lifetime expires, without getting requested Chunk, the consumer resends the Interest packet.
- TCP sliding window is adjusted based on the packet loss and RTO, which are the signal of congestion, i.e., TCP reduces window size when congestion is happening. On the other hand, FDCC proactively prevents congestion through adjusting sliding window, where the window size is reduced each time congestion alert messages are received.

C. COMMUNICATION BETWEEN CMTB AND FDCC

In Sections IV-A and IV-B, CMTB (which uses M/M/1/B queuing model) and FDCC (which uses sliding window) were discussed in details; where CMTB is implemented in the intermediate node(s), while FDCC is implemented on consumer's side. Moreover, the interoperability of queuing based model and sliding window based model was analyzed in details in [33]. In this subsection, we discuss the communication (about the congestion) between CMTB and FDCC through the use of RSR message.

1) REDUCE SENDING RATE (RSR) MESSAGE EXCHANGE

RSR Message is a special kind of message that each intermediate node sends when all its buffers reach the threshold T_k. The node returns the RSR message to its downstream node so that the downstream node could adjust the traffic. When the downstream node is the intermediate node, it uses CMTB. However, when the downstream node is the consumer node, it uses FDCC.

In order to ensure the originality of the CCN packet structure, where it has two types of packets: the Interest Packet and the Data packet. Another special packet for RSR has not been created. RSR is an auto-generated Interest packet when intermediate node's buffers reach thresholds, it has the same format of Interest packet. Therefore, RSR is allowed to be sent out to the downstream node only at a one-hop distance, and then gets dropped. In other words, RSR doesn't need always to be forwarded to the downstream node. For generating and sending RSR, the node uses the incoming face(s) available in PIT, from which the Interests that causes buffer overflow are received.

The downstream node differentiates RSR packet with other Interest packets based on prefix name, where prefix name/RSR is only used for RSR message. In Algorithm 3, upon reception of that special Interest packet with prefix name/RSR, the Intermediate node calls CMTB at line 5 for adjusting the traffic rate on outgoing faces. On the other hand, at line 8, consumer node uses FDCC to adjust the sending rate. The RSR message reaches the consumer node, when all intermediate nodes fail to adjust to the traffic rate. In other words, when the intermediate nodes are able to adjust the Interest traffic rate, the RSR does not reach the consumer node.

Algorithm 3 Reduce Sending Rate.

1: **Precondition:** A node receives Interest i;
2: At Interest i reception (prefix, inface $k \in \mathcal{K}_v$);
3: **if** Interest i's prefix = /RSR **then**
4:     **if** $v \in \mathcal{V}$ is an intermediate node **then**
5:         Use CMTB for new incoming packet i;
6:         Drop Interest i;
7:     **else** {Hint: $v \in \mathcal{V}$ is a consumer node}
8:         Use FDCC to adjust outgoing Interest i;
9:         Drop Interest i;
10:     **end if**
11: **else** {Hint: Interest i's prefix $\neq$ /RSR}
12:     Follow the normal CCN procedure;
13: **end if**

The objective of sending the RSR message to the consumer node when all intermediate nodes have failed to handle the congestion is because the consumer is the only one who has the last available solution for reducing the Interest traffic w_m. Without controlling how the consumer generates Interest traffic, congestion in CCN network is inevitable.

V. PERFORMANCE EVALUATION

In this section, we present in detail the performance analysis of our proposal. We use ns-3 based simulator, namely the ndnSIM 2.1 [34], [35] to analyze the performance of NCFCC, and compare it with other similar proposals in the literature.

A. CONGESTION CONTROL MECHANISMS UNDER COMPARISON

- MIRCC: We evaluate our proposed congestion control mechanism by comparing it with Multipath-aware ICN Rate-based Congestion Control (MIRCC) [12], a rate-based multipath-aware congestion control scheme for ICN.
- Hop by Hop Interest shaping mechanism for CCN (HoBHIS): In HoBHIS [10], for preventing congestion, it computes available capacity in each CCN router in order to regulate Interest rates.
- Popularity-based Congestion Control (PbCC) [11]: PbCC prevents congestion through taking into account of CCN caching, link bandwidth, Interest and Data packets received in order to adjust the outgoing Interest rate that needs to be forwarded to the upstream link(s).

B. EXPERIMENTAL SETUP

For the experimental setup, which is classified into two categories: (1) Single path scenario, and (2) Multipath scenarios, we use three topologies with total workload Γ of 100 MB per each topology.

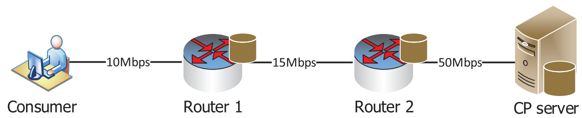

FIGURE 5. Scenario 1: Single path topology.

1) SINGLE PATH SCENARIO

In the first simulation scenario described in Fig. 5, we have one consumer node, two intermediate routers, and one CP. There are three consumer applications installed on the consumer node. The first application requests the content through the use of prefix name $/com/ccn$. The second application requests the content through the use of prefix name $/com/news$, while the third application requests the content through the use of prefix name $/com/video$.

2) MULTIPATH SCENARIOS

In the second simulation scenario, we use our system model described in Fig. 2, with ten consumers, four intermediate routers, and three CPs. The first five consumers are connected to router 1, and other five consumers are connected to the router 2. The first three consumers request content hosted in CP 1 through the use of prefix name $/com/ccn$, consumers $4-7$ request the content hosted in CP 2 through the use of prefix name $/com/news$, while consumers $8-10$ request the content hosted in CP 3 through the use of prefix name $/com/video$. Table 2 shows the details of the rest of the parameters of scenario 2.

We have conducted additional simulations of multipath in realistic network topology (simulation scenario 3), namely GEANT topology [36], which is presented in Fig. 6. GEANT topology connects many research centers and universities in Europe through the use of 22 routers in its core network. We extend the topology with one CP on the core network, and 20 consumers (c1-c20). Table 3 shows the details of the rest of the parameters of scenario 3.

3) INTEREST TRAFFIC GENERATION

In both scenarios (single path scenario and multipath scenarios), we use FDCC for generating, and adjusting Interest

TABLE 2. Scenario 2 topology.

Parameter	Value
Topology	System model
V	4 CCN routers
S	10 consumers (Src01-Src10)
P	3 CP servers
N	3 flows
B_k	100 packets
T_v	50 packets
l_i	1040 Bytes
C_v	1000 Chunks in each router v
Γ	100 MB
w_m	10 Interest packets
q	$1/2$ (on RSR), $1/3$ (on Interest lifetime expiration)
C_j	10 Mbps for each link between consumers and access routers $(1, 2)$, 15 Mbps for each link between routers $1-3, 2-3$, and $2-4$ (denoted v in Fig. 2). 50 Mbps for each link between router $3-CP$ $(1, 2, 3)$ and router $4-CP$ $(1, 2, 3)$
Simulation time	100 Seconds

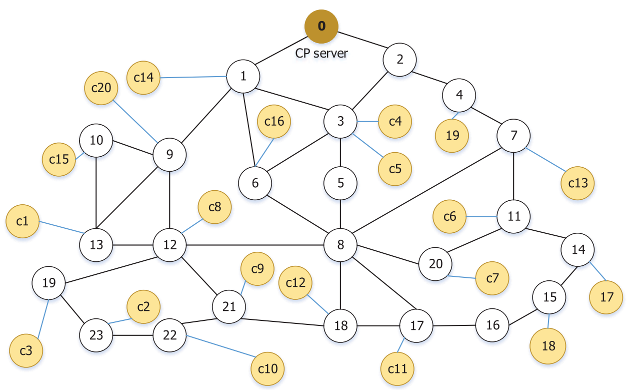

FIGURE 6. Scenario 3: GRANT2 Network Topology with 20 consumers.

TABLE 3. Scenario 3 topology.

Parameter	Value
Topology	GEANT 2
V	20 CCN routers
S	20 consumers
P	1 CP servers
N	3 flows
B_k	100 packets
T_v	50 packets
l_i	1040 Bytes
C_v	1000 Chunks in each router v
Γ	100 MB
w_m	10 Interest packets
q	$1/2$ (on RSR), $1/3$ (on Interest lifetime expiration)
C_j	100 Mbps
Simulation time	100 Seconds

packets on consumer nodes, where the initial window size was set to $w_m = 10$ Interest packets. However, as described in FDCC, due to the received Data packet, RSR, and RTO, consumer nodes can increase or reduce w_m. On the other hand, other congestion control mechanisms under comparison use different methods for generating Interest packets, which are described as follows:

- In HoBHIS, ConsumerCbr [35] was used for generating Interest packets, which is based on pre-defined parameters such as frequency, traffic rate, etc. Moreover, in this mechanism, the frequency was set to 100 Interest packets per second.
- In PbCC, consumer nodes generate Interest packets through the use of ConsumerZipfMandelbrot [35]. ConsumerZipfMandelbrot works as ConsumerCbr, in which Interest traffic is generated based on pre-defined frequency. In PbCC, the frequency was set to 100 Interest packets per second, while both ZipfMandelbrot and power parameters were set to 0.6 (default value in PbCC proposal).
- In MIRCC, the initial Interest traffic generation was set to 100 Interest packets per second, then consumer nodes adjust Interest traffic rate for each outgoing link based on returning Data rate.

C. PERFORMANCE METRICS

1) THROUGHPUT

Throughput is defined as a measurement of how many units of information network can handle for a given period of time. In the other words, network throughput is the amount of Data moved successfully from one network node to another in a given time period [37]. In this paper, we used an estimated rate within last averaging period of time (kilobits per second), which is based on Exponentially Weighted Moving Average (EWMA) [35].

2) PACKET DELAY

The Interest packet generated by consumer node goes through a series of nodes and ends its journey in another node, which has requested Chunk in its CS, where that node returns the Chunk in the reverse path of Interest packet. Moreover, both Interest and Chunk packets experienced several types of the delays at each node along the transmission path. These delays are processing delay, queuing delay, transmission delay, and propagation delay, where the total of these delays is known as total nodal delay [38]. Suppose that there are $N - 1$ nodes between the consumer node and the node that has the requested Chunk in its content store. The total nodal delay accumulated for N nodes, giving us packet delay, i.e., we use packet delay as the delay between issuing Interest packet, and receiving corresponding Data/Chunk.

3) INTEREST RETRANSMISSION RATE

In Section IV-B, we discussed Interest lifetime and RTO_i, where on expiration of Interest lifetime or RTO_i, the consumer node needs to retransmit Interest packet. To compare our proposal with other mechanisms and check the effectiveness of our proposal, in our simulation, we trace the number of retransmitted Interest packets.

D. SIMULATION RESULTS

1) SINGLE PATH

Figs. 7, 8 and 9 show the simulation results of our algorithms in the single path scenario, where FDCC algorithm is

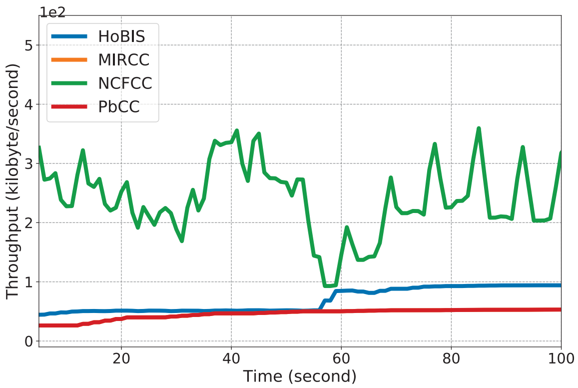

FIGURE 7. Scenario 1: Time-based throughput.

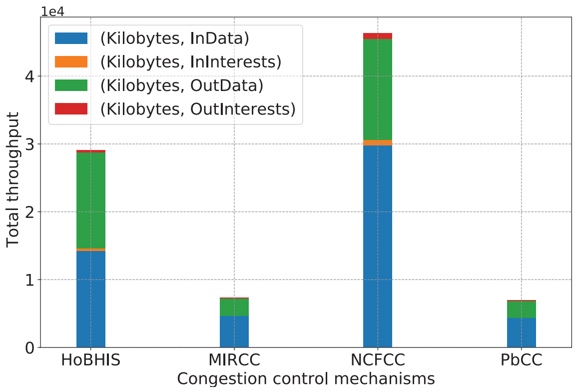

FIGURE 8. Scenario 1: Total throughput (kilobytes/100 seconds) comparison.

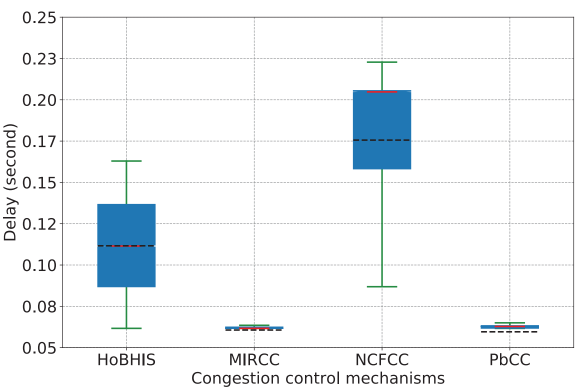

FIGURE 9. Scenario 1: Delay comparison.

deployed in one consumer node, while the CMTB is deployed in two intermediate routers.

In Fig. 7, we compare NCFCC with other well-known three congestion control mechanisms, namely HoBHIS, PbCC and MIRCC. The NCFCC achieves a higher throughput than others, while PbCC and MIRCC have almost the same throughput. From the beginning, we start the simulation with single flow through the use of consumer application 1. Then at 50, we start two more flows through the use of consumer applications 2 and 3. In both single flow and multi-flows, the NCFCC is characterized by high performance.

Fig. 8 shows the total throughput comparison, i.e., the total amount of Interest and Data packets moved successfully from one node to another during 100 seconds of simulation. We remind that, in CCN, there are two types of packets: Interest Packets and Data packets. Therefore, total throughput includes InInterests, OutInterests, InData and OutData. As defined in [39], InInterests measure total amount of incoming Interest packets in terms of kilobits per second, while OutInterests measures outgoing Interest packets. Moreover, InData represents total amount of incoming Data packets, while OutData represents the outgoing Data packets. In this simulation setup, the consumer nodes participate in content distribution through sharing cached contents. However, the contents that are not cached on consumer nodes, can be retrieved either from the caches implemented in the transmission path or from CP server.

Network throughput and packet delay are the two most important metrics for evaluating the performance of congestion control mechanism. Therefore, Fig. 9 represents the delay experienced by packets (both Interest and Data packets). In this figure, solid red lines represent median, while dashed black lines represent arithmetic mean. NCFCC has a higher delay than the others, due to the fact that NCFCC has high throughput, i.e., NCFCC sends and receives more packets than other congestion control mechanisms under comparison. In addition, on reception of congestion alert message (RSR) from upstream node $v \in \mathcal{V}$, due to the presence of single path, CMTB cannot allocate the new incoming Interest packets to another outgoing face(s). On the other hand, HoBHIS has a high delay due to the traffic shaping delay in which different Interest packets have to meet, when buffer threshold is reached. PbCC and MIRCC experienced almost the same delay. MRCC and PbCC have lower delay than the others, due to the fact that MRCC and PbCC has lower throughput, i.e., they send and receive less packets in the network than other congestion control mechanisms. Furthermore, in the single path scenario, there is no packet loss and retransmission in all congestion control mechanisms that are under comparison in this scenario.

2) MULTIPATH

In simulation scenario 2 for multipath, we use our system model. There are three bottleneck links between routers 1-3, routers 2-3, and between routers 2-4. Our FDCC algorithm is deployed in 10 consumer nodes (Src01-Src10), while the CMTB is deployed in 4 intermediate routers ($R1$, $R2$, $R3$, $R4$).

Figs. 10, 11, 12, 13, 14, and 15, show the simulation results of scenario 2. In Fig. 10, we compare our congestion control mechanism with other three congestion control mechanisms (MIRCC, HoBHIS, and PbCC). During 100 seconds of simulation, we create a link failure between router $R2$ and router $R4$ for a period of 20 seconds starting from 20*th* second and ending at 40*th* second. In this simulation setup, Fig. 10 shows that the NCFCC achieves higher throughput than others. Furthermore, Fig. 11 shows the total throughput per each

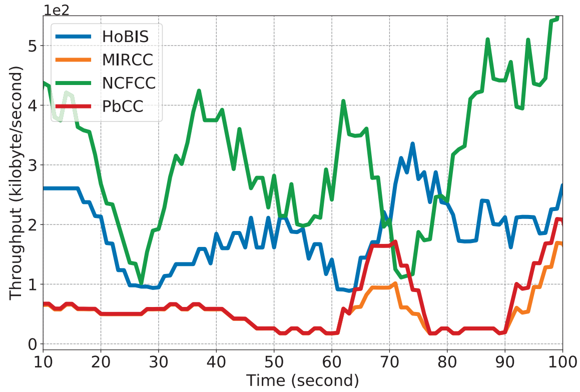

FIGURE 10. Scenario 2: Time-based throughput.

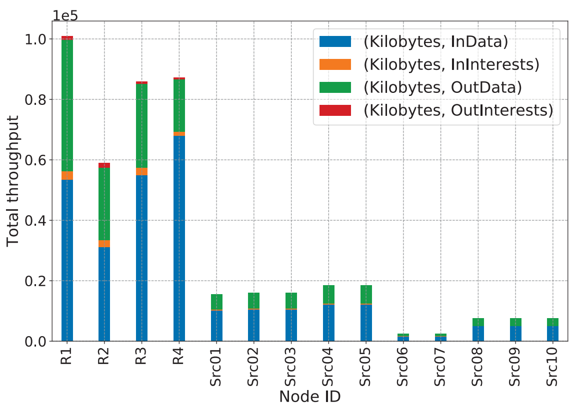

FIGURE 11. Scenario 2: Total throughput (kilobytes/100 seconds) of each node (NCFCC).

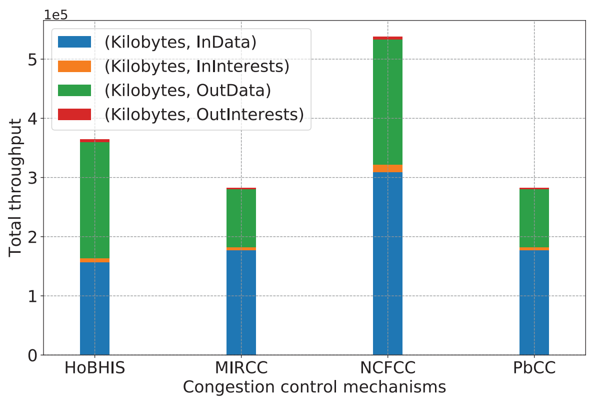

FIGURE 12. Scenario 2: Total throughput (kilobytes/100 seconds) comparison.

node for NCFCC, while the Fig. 12 shows consolidated total throughput comparison.

Fig. 13 shows the delay experienced by packets, where NCFCC has lower delay than HoBHIS. The simulation results in this figure show that NCFCC utilizes multipath effectively, which contributes to the reduction of the delay experienced by packets and the increase of network throughput.

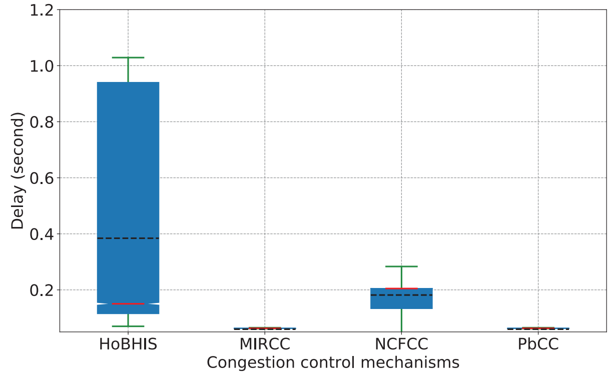

FIGURE 13. Scenario 2: Delay comparison.

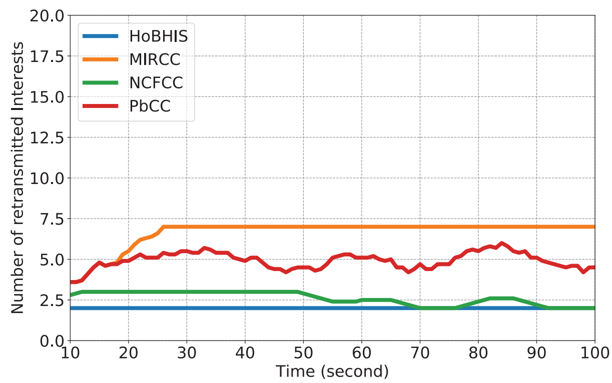

FIGURE 14. Scenario 2: Interest retransmission.

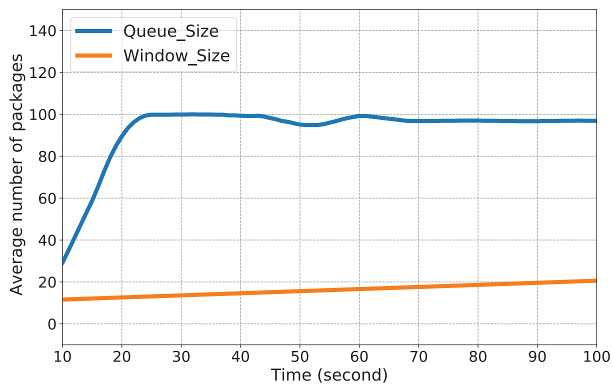

FIGURE 15. Scenario 2: Queue size and window size.

Fig. 14 shows the average number of retransmitted Interest packets due to the link failure. MRCC and PbCC have experienced higher packet retransmissions than NCFCC and HoBHIS. In this scenario, HoBHIS has not experienced retransmission oscillations than the others.

Fig. 15 shows the average queue and window size variations for NCFCC, where the maximum buffer size per each face is limited to 100 packets. During 100 seconds of simulation, all routers prevent queue overflow through the use of CMTB algorithm and RSR notification.

Figs. 16 and 17 show the simulation results for GEANT2 network topology (simulation scenario 3). Our FDCC

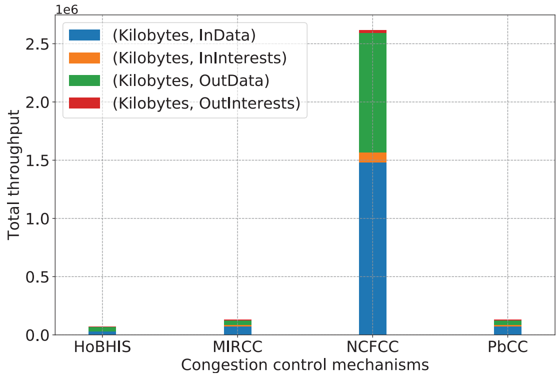

FIGURE 16. Scenario 3: Total throughput (kilobytes/100 seconds) comparison.

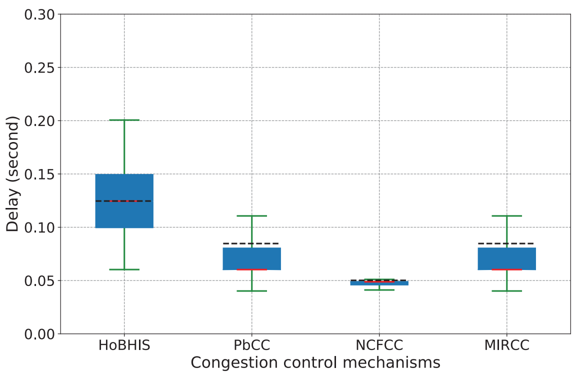

FIGURE 17. Scenario 3: Delay comparison.

algorithm is deployed in 20 consumer nodes, while the CMTB is deployed in 23 intermediate routers.

Fig. 16 shows that NCFCC has higher throughput, while Fig. 17 shows that NCFCC has lower delay than others due to the following reasons:

- In this scenario, the topology has high-bandwidth (each link has capacity $C_j = 100$ Mbps) and multiple paths to reach the CP. In this setting, there is no bottleneck link and link failures.
- In NCFCC, we implemented FDCC on the consumer node. For each Data packet received, the consumer node keeps increasing window size/number of outstanding Interest packets. Therefore, consumer nodes with FDCC can have large number of outstanding Interest packets (up to $w_m = 1311$ Interest packets) while waiting for the Data packets. Since Interest packets are much smaller than the Data packets [40], having large number of outstanding Interest packets have negligible effect on uplink congestion.
- In NCFCC, when intermediate node receives RSR message (messages generated by CMTB) from upstream node $v \in \mathcal{V}$, it allocates new incoming Interest packets to other outgoing faces associated to other nodes rather than v, which results in effective utilization of multiple paths and thus reducing packet delay. In other words, as the number of the paths increases, the packet delay decreases.
- On the other hand, MIRCC, HoBHIS, and PbCC have lower throughputs and higher delays than NCFCC, even when the network has high bandwidth connection and multiple paths, because each Interest packet submitted takes a complete VRTT to receive the requested Data packet and the consumer can only send the next Interest after receiving the Data packet. This results in higher delay and link underutilization [41] [42].
- HoBHIS has higher delay than others due to its packet shaping delay that Interest packets have to meet when the buffer threshold is reached. Furthermore, HoBHIS has lower throughputs than others due to its Interest traffic generation method at consumer nodes, where the Interest packets are generated at a constant rate (100 Interest packets per second) without considering the network condition and link capacity.

Therefore, due to the high-bandwidth, multiple paths, and absence of bottleneck links, in all congestion control mechanisms used in this scenario, there is no packet loss and retransmission.

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we presented Novel Cooperative and Fully-distributed Congestion Control (NCFCC) mechanism for CCN, which prevents congestion before it happens. NCFCC combines inseparable hop-by-hop CMTB, which is a modified version of the Dynamic Token Bucket, and receiver-based congestion control FDCC (as modified version of Additive-Increase/Multiplicative-Decrease) into one hybrid congestion control mechanism, which increases network throughput and reduces delay experienced by packets. In the absence of FDCC (through the use of CMTB only), congestion is inevitable in intermediate nodes, thus consumers continue to generate more traffic. However, in combining both algorithms, when all intermediate nodes fail to prevent congestion through the use of CMTB, CMTB calls (through sending RSR messages) FDCC for reducing traffic rate. On receiving RSR message, FDCC takes the lead by adjusting the sending rate in consumer nodes. Our simulation results show that our proposed scheme, over different types of simulation environments, achieves significant improvements with higher throughput than other similar proposals in the literature. In the future, we aim to extend our proposal to further minimizing delay, and examine in details the complexity and scalability of our algorithms.

REFERENCES

[1] S. Ullah, K. Thar, and C. S. Hong, "Management of scalable video streaming in information centric networking," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21519–21546, 2017.

[2] *Cisco Visual Networking Index: Forecast and Methodology, 2015–2020*. Accessed: Aug. 15, 2017. [Online]. Available: http://www.communicationstoday.co.in/reports/8556-ciscovisual-networking-index-forecast-and-methodology-2015-2020

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Experim. Technol.*, 2009, pp. 1–12.

[5] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2014.

[6] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Comput. Commun.*, vol. 36, no. 7, pp. 779–791, 2013.

[7] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, "Congestion control in named data networking—A survey," *Comput. Commun.*, vol. 86, pp. 1–11, Jul. 2016.

[8] L. Saino, C. Cocora, and G. Pavlou, "CCTCP: A scalable receiver-driven congestion control protocol for content centric networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 3775–3780.

[9] C. Stais, G. Xylomenos, and A. Voulimeneas, "A reliable multicast transport protocol for information-centric networks," *J. Netw. Comput. Appl.*, vol. 50, pp. 92–100, Apr. 2015.

[10] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for CCN communications," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Mar. 2012, pp. 322–327.

[11] H. Park, H. Jang, and T. Kwon, "Popularity-based congestion control in named data networking," in *Proc. 6th Int. Conf Ubiquitous Future Netw. (ICUFN)*, 2014, pp. 166–171.

[12] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "MIRCC: Multipath-aware ICN rate-based congestion control," in *Proc. Conf. 3rd ACM Conf. Inf.-Centric Netw.*, 2016, pp. 1–10.

[13] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," in *Proc. 2nd Ed. ICN Workshop Inf.-Centric Netw.*, 2012, pp. 37–42.

[14] G. Carofiglio, M. Gallo, and L. Muscariello, "Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation," *Comput. Netw.*, vol. 110, pp. 104–117, Dec. 2016.

[15] S. Oueslati, J. Roberts, and N. Sbihi, "Flow-aware traffic control for a content-centric network," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2417–2425.

[16] A. Karami, "Accpndn: Adaptive congestion control protocol in named data networking by learning capacities using optimized time-lagged feedforward neural network," *J. Netw. Comput. Appl.*, vol. 56, pp. 1–18, Oct. 2015.

[17] G. Appenzeller, I. Keslassy, and N. McKeown, *Sizing Router Buffers*, vol. 34. Pennsylvania, PA, USA: ACM, 2004.

[18] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," Internet Eng. Task Force/Netw. Working Group, Fremont, CA, USA, Tech. Rep. RFC 568, 2009.

[19] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," Internet Eng. Task Force, Fremont, CA, USA, Tech. Rep. RFC 2001, 1997.

[20] Y. Kim, Y. Kim, J. Bi, and I. Yeom, "Differentiated forwarding and caching in named-data networking," *J. Netw. Comput. Appl.*, vol. 60, pp. 155–169, Jan. 2016.

[21] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proc. 23rd Int. Teletraffic Congr.*, 2011, pp. 111–118.

[22] A. Ndikumana *et al.*, "Network-assisted congestion control for information centric networking," in *Proc. 17th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Aug. 2015, pp. 464–467.

[23] D. Wischik and N. McKeown, "Part I: Buffer sizes for core routers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 75–78, 2005.

[24] J. Kidambi, D. Ghosal, and B. Mukherjee, "Dynamic token bucket (DTB): A fair bandwidth allocation algorithm for high-speed networks," in *Proc. 8th Int. Conf. Comput. Commun. Netw.*, 1999, pp. 24–29.

[25] O. M. Heckmann, *The Competitive Internet Service Provider: Network Architecture, Interconnection, Traffic Engineering and Network Design*. Hoboken, NJ, USA: Wiley, 2007.

[26] R. S. Prasad, C. Dovrolis, and M. Thottan, "Router buffer sizing for TCP traffic and the role of the output/input capacity ratio," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1645–1658, Oct. 2009.

[27] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Comput. Netw.*, vol. 57, no. 16, pp. 3128–3141, 2013.

[28] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Netw. ISDN Syst.*, vol. 17, no. 1, pp. 1–14, 1989.

[29] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing TCP's retransmission timer," Internet Eng. Task Force, Fremont, CA, USA, Tech. Rep. RFC 6298, 2011.

[30] M. Chiang, *Networked Life: 20 Questions and Answers*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[31] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[32] *NDN Packet Format Specification 0.2-2 Documentation*. Accessed: Aug. 15, 2017. [Online]. Available: https://named-data.net/doc/ndn-tlv/interest.html

[33] G. Stylianou, J. Nottingham, L. Mahalingam, M. Khambatti, and S. Gnanaprakasam, "Analysis of the sliding window protocol over multiple hops for fault-tolerant networks," College Inf. Sci. Technol., Pennsylvania State Univ., State College, PA, USA, 2010.

[34] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Los Angeles, CA, USA, Tech. Rep. NDN-0005, Oct. 2012. [Online]. Available: http://named-data.net/techreports.html

[35] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2: An updated NDN simulator for NS-3," NDN, Los Angeles, CA, USA, Tech. Rep. NDN-0028, Nov. 2016.

[36] *Geant Topology Map*. [Online]. Available: http://www.geant.org/Networks/Pan-European_network/Pages/GEANT_topology_map.aspx

[37] S. C. Satapathy and A. Joshi, *2nd International Conference on Information and Communication Technology for Intelligent Systems*, vol. 2. Berlin, Germany: Springer, 2017.

[38] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*. Reading, MA, USA: Addison-Wesley, 2007.

[39] *ndnSIM: Obtaining Metrics*. Accessed: Aug. 15, 2017. [Online]. Available: http://ndnsim.net/2.1/metric.html

[40] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, D. Kim, and M. Gerla, "CONET: Controlled data packets propagation in vehicular named data networks," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2016, pp. 620–625.

[41] A. Ndikumana, S. Ullah, and C. S. Hong, "Scalable aggregation-based packet forwarding in content centric networking," in *Proc. 18th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Oct. 2016, pp. 1–4.

[42] D. Byun, B.-J. Lee, and M.-W. Jang, "Adaptive flow control via interest aggregation in CCN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 3738–3742.

ANSELME NDIKUMANA received the B.S. degree in computer science from the National University of Rwanda in 2007. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. His professional experience includes a Chief Information Officer, a System Analyst, and a Database Administrator at Rwanda Utilities Regulatory Authority from 2008 to 2014. His research interest includes congestion control, multiaccess edge computing, and in-network caching.

SAEED ULLAH received the B.S. degree in information technology from IBMS/CS, Peshawar, and the M.S. degree in information technology from the National University of Sciences and Technology in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. His research interest includes multimedia communication, scalable video streaming, routing and in-network caching, and future internet.

KYI THAR received the bachelor's degree in computer technology from the University of Computer Studies at Yangon, Yangon, Myanmar, in 2007. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, South Korea, for which he received the scholarship for his graduate study in 2012. His research interest includes name-based routing, in-network caching, multimedia communication, scalable video streaming, wireless network virtualization, and future internet.

NGUYEN H. TRAN (S'10–M'11) received the B.S. degree in electrical and computer engineering from the Ho Chi Minh City University of Technology in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University in 2011. Since 2012, he has been an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interest is applying analytic techniques of optimization, game theory, and stochastic modeling to cutting-edge applications, such as cloud and mobile-edge computing, datacenters, heterogeneous wireless networks, and big data for networks. He received the Best KHU Thesis Award in engineering in 2011 and several best paper awards, including the IEEE ICC 2016, the APNOMS 2016, and the IEEE ICCS 2016. He is an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING.

BANG JU PARK received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, South Korea, in 1986 and 1988, respectively, and the Ph.D. degree in radio engineering from the Korea Institute of Science and Technology, Kyung Hee University, South Korea, in 2006. From 1988 to 2012, he was a Science Reporter with JoongAng Daily, South Korea. From 2008 to 2012, he was a Visiting Professor with the College of Bio-Nano Engineering, Gachon University, South Korea. In 2013, he moved to Gachon University where he is currently a Full Professor with the Department of Electronic Engineering and the Director of the Institute of Bio and Mechatronics. His research interests include future internet and radio-manipulation for biomedical instrumental applications.

CHOONG SEON HONG received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University in 1997. In 1988, he joined Korea Telecom (KT), where he was involved in broadband networks as a member of Technical Staff. In 1993, he joined Keio University, Japan. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and as the Director of Networking Research Team, until 1999. Since 1999, he has been a Professor with the Department of Computer Engineering, Kyung Hee University. His research interests include future internet, ad hoc networks, network management, and network security. He is a member of the ACM, the IEICE, the IPSJ, the KIISE, the KICS, the KIPS, and the OSIA. He has served as the General Chair, the TPC Chair/Member, or an Organizing Committee Member for international conferences, such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT, and ICOIN. He is currently an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the *International Journal of Network Management*, and the IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS, and an Associate Technical Editor of the *IEEE Communications Magazine*.