Adaptive Multi-Source Multi-Path Congestion Control for Named Data Networking

Jiayu Yang, *Member, IEEE*, Yuxin Chen, *Graduate Student Member, IEEE*, Kaiping Xue, *Senior Member, IEEE*, Jiangping Han, *Member, IEEE*, Jian Li, *Senior Member, IEEE*, Ruidong Li, *Senior Member, IEEE*, Qibin Sun, *Fellow, IEEE*, and Jun Lu

***Abstract*— Named Data Networking (NDN), with a receiver-driven connectionless communication paradigm, naturally supports content delivery from multiple sources via multiple paths. In a dynamic environment, sources and paths may change unexpectedly and are uncontrollable for consumer, which requires flexible rate control and real-time multi-path management, still lacking investigations. To address this issue, we propose an Adaptive Multi-source Multi-path Congestion Control (AMM-CC) scheme based on online learning. AMM-CC explores source/path distribution with continuous micro-experiments and abstracts the empirically experienced performance by meticulously designed two-level utility functions. Specifically, AMM-CC enables each consumer to optimize a local transmission-level utility function that fuses multi-source characteristics, including congestion level and source weights. Then, a sub-gradient descent method is designed to adjust transmission rate adaptively and achieve fine-grained control. Moreover, AMM-CC coordinates consumer with the forwarding module to ensure efficient and on-time multi-path management. It enables consumer to determine congestion gap among multiple paths by a path-level utility that sensitively captures changes and congestion on each path. Then, consumer further notifies the forwarding module in achieving precise traffic transferring. We conducted comprehensive evaluations in dynamic scenario with various content distribution using the NDN simulator, ndnSIM. The evaluation results demonstrate that AMM-CC can adapt to flexible content acquisition from multi-sources and significantly improve bandwidth utilization of multi-path compared with state-of-the-art schemes.**

***Index Terms*— Named data networking, congestion control, online learning.**

I. Introduction

TO SUIT new requirements of network transmission in terms of mobility support [1] and efficient content distribution [2], the considerable research attention has been devoted to revolutionary network architectures. Therefore, various Information-Centric Networking (ICN) architectures aiming at efficiently retrieving data have been proposed [3]. Among them, the Named Data Networking (NDN) [4], is a typical and promising paradigm for its simple and effective transmission mode. With NDN, the network architecture has been redesigned and the end-to-end transmission mode of the traditional network has been changed into a name-based connectionless transmission. The NDN network supports pull-based content retrieval driven by consumers' requests. It breaks the binding between content and address with name-based routing and hop-by-hop packet forwarding, enhancing flexible mobility management. Moreover, the presence of embedded in-network caches enables the reuse of previously transmitted content, further improving network resource utilization.

The communication paradigm of NDN natively supports flexible content transmissions from multi-source via multi-path in a dynamic fashion, intrinsically coupled with in-network caching. In a NDN network, contents are cached in a distributed manner throughout the network, and there are multiple content sources that respond to consumer requests, including an original content producer and in-network caches that cache a content copy. Usually, content requests sent by consumers are dynamically dispatched by taking on-the-fly forwarding decisions at intermediate routers. The request (interest packet) is forwarded hop-by-hop until the content is hit. Then, data packets follow the reverse paths of the requests to return to consumers. Since transmitted content may be cached in routers close to consumer, and cached contents may be replaced by other content due to limited buffer size, the available content sources and alternative paths may change dynamically during content transmissions.

In a dynamic environment, congestion control, a vital element to guarantee efficient transmissions in a NDN networks, faces challenges of multi-source and multi-path utilization. Firstly, *adaptive rate control*: as the cached content of routers is unpredictable to consumers, consumer is unknown to multi-source and multi-path. Thus, it regards all of its available sources as one entity and maintains a whole transmission rate to implement rate adjustments. In a dynamic environment, transmission sources may constantly change, including their proportion to the whole transmission and the status of their transmission paths. It is a tricky task for consumers to adjust the transmission rate to adapt to multi-source states and

Manuscript received 25 September 2023; revised 10 June 2024 and 7 August 2024; accepted 16 August 2024; approved by IEEE/ACM Transactions on Networking Editor S. Ioannidis. Date of publication 27 August 2024; date of current version 19 December 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62302472 and Grant 62372425, in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant Y202093, and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding authors: Kaiping Xue; Jiangping Han.)*

Jiayu Yang, Yuxin Chen, Kaiping Xue, Jiangping Han, Jian Li, Qibin Sun, and Jun Lu are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: kpxue@ustc.edu.cn; jphan@ustc.edu.cn).

Ruidong Li is with the College of Science and Engineering, Kanazawa University, Kakumamachi, Kanazawa 920-1192, Japan.

Digital Object Identifier 10.1109/TNET.2024.3447467

avoid congestion. Secondly, *efficient multi-path utilization*: unlike Multi-Path Transmission Control Protocol (MPTCP) in traditional Internet environments and data center networks [5], [6], [7], [8], [9], which enables efficient load balancing over its static multi-path, NDN with a connectionless transmission mode lacks multi-path management strategies at the consumer. Instead, NDN relies on a forwarding module that selects forwarding interfaces hop-by-hop to decide how traffic is distributed across different paths in the network. However, the forwarding module is transparent to the multi-path composition at consumer. Thus, it guarantees transmission efficiency by balancing the traffic of all links. However, it is tricky and cost to achieve real-time balancing in a dynamic environment. On the one side, with time-varying path states, the optimal forwarding with perfect load balancing obtained by solving the modeled multi-commodity flow problem [10] has been proven to be NP-hard and time-consuming [11].

Until now, there has been a large number of congestion control schemes and forwarding strategies to pursue the efficient utilization of multiple sources and multiple paths in NDN networks. Some schemes decouple congestion control and forwarding modules [12], [13]. They provide consumer-side rate adjustment algorithms and totally rely on the existing forwarding modules [14], [15], [16] to manage traffic on multiple paths. Some schemes jointly consider rate control and forwarding mechanism to provide different ways to improve multi-path utilization [10], [17], [18], including designing an optimal forwarding strategy, providing a traffic transfer scheme with congestion notification, and proposing a newly designed path-specified data acquisition mode. All of them play a positive role in improving the transmission efficiency of NDN networks. However, all of them are suitable for stable transmissions and suffer from low transmission efficiency in a dynamic environment.

A more flexible strategy is required to enable efficient multi-source multi-path control in a dynamic NDN network, where consumer faces changing sources and paths. Specifically, in a dynamic environment, sources may occupy a varying proportion of the whole transmissions and paths also suffer from varying congestion status. In this case, a predefined rate adjustment scheme with static actions is powerless to achieve effective control. For example, if a consumer defines to halve the window after sensing a packet loss, it avoids congestion diffusion when all sources are congested. However, when there is only one congested source transferring small amounts of data, it causes excessive traffic decrease for non-congested ones.

To avoid hard-wired mapping of packet-level events to static predefined actions, we leverage emerging online learning techniques to achieve adaptive control. It achieves fine-grained control under dynamic environments through real-time detection and online decision, maintaining lower overhead and being well-suited for deployment on consumers. However, there are still problems to be solved for achieving the desired transmission efficiency. *First, the appropriate reaction design to the congested source is necessary:* Facing with congestion alert of a source, inadequate response may deteriorate congestion, while overreaction results in underutilization of network resources. *Second, the scheme should be sensitive to changes, while ensuring stable control of overall transmissions:* Facing with dynamic multi-source multi-path transmission, it is necessary to timely capture changes of their transmission status. However, it is also essential to ensure the stability and convergence of the overall transmission rate.

To address the above problems, we propose an Adaptive Multi-source Multi-path Congestion Control (AMM-CC) scheme based on online learning. Specifically, AMM-CC leverages online convex optimization for adaptive control under dynamic environments. It continuously explores source and path distribution and then treats them differently based on their influence on the overall transmission with the well designed utility functions. Moreover, we provide two levels of control to ensure the stability of the overall transmission rate and guarantee sensitive response to congested paths. To be specific, AMM-CC includes two modules to achieve transmission level and path level control: an Adaptive Rate Adjustment (ARA) strategy and a Lightweight Multi-path Balancer (LMB). The former focuses on the network's overall performance, guaranteeing smooth rate adjustment for the consumer. It also ensures convergence for multi-source and multi-path rate control. The latter pays attention to real-time status of single-path, which responds sensitively to congestion gaps among paths, further improving resource utilization of multiple paths.

The main contributions of this paper are summarized as follows:

- We provide an adaptive congestion control scheme, called AMM-CC, for dynamic NDN networks with changing multi-source and multi-path. AMM-CC delves into the intricate control of the whole rate and multi-path traffic management, affording a dual-layered control mechanism.
- We provide an ARA module for rate adjustment under changing multi-source based on online convex optimization. By a meticulously designed transmission-level utility function, ARA artfully amalgamates the contributions of each individual source. It continuously observes empirically experienced performance and achieves fine-grained control over transmission rate with the sub-gradient descent method.
- We introduce a lightweight multi-path balancer for timely optimizing multi-path utilization. By a path-level utility function, consumer examines congestion gaps across available paths and gauges excessive traffic on overwhelmed paths. Then, it coordinates with in-network forwarding modules through notification information in Interest packets, achieving accurate traffic transfer.
- We theoretically analyze the convergence and stability of AMM-CC, and conduct comprehensive evaluations with various path states and content distribution in ndnSIM, a NDN simulator. Our evaluation results reveal that AMM-CC can improve goodput by up to 102.62% in dynamic cache hit ratios, without compromising the utilization of newly emerging content sources.

The rest of this paper is organized as follows. In Section II and Section III, we present the background of NDN networks and related works. Then, the problem formulation and design details of AMM-CC scheme are illustrated in Section IV and Section V, respectively. Subsequently, we provide a theoretical analysis and performance evaluation in Section VI and Section VII. Finally, we conclude our work in Section VIII.

II. Background of NDN Networks

NDN, first proposed by Jacobson [19] and later supported as one of NSF FIA projects [4], is one of the most popular paradigms of the next-generation networks. It provides a novel "content" oriented transmission paradigm, where consumers obtain content by sending requests with specifying content names. Generally, there are two kinds of packets in NDN networks: Interest and Data packets, and each router maintains three types of data structures to support content delivery: Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT). NDN follows "pull-based" content acquisition pattern, where Interest packets are forwarded hop-by-hop to find the required content, strictly obeying the "one-interest-one-data" principle. Specifically, the data acquisition process is as follows: A consumer sends out an Interest packet carrying data name. This packet is forwarded hop-by-hop by routers, which checks the its information tables to tackle the packet. Firstly, router checks its CS to search matching data. If there is a temporary copy of the requested content in CS, the data packet is sent back to consumer from which Interest is received. Otherwise, router looks up the name in its PIT table to check if there is already a recorded Interest with the same data name. If it exists, the incoming interface of this Interest is simply added in that PIT entry; otherwise, router forwards the Interest packet based on the information in the FIB as well as its adaptive forwarding strategy, which decides to send the Interest to one or multiple interfaces. The Interest packet is forwarded hop-by-hop until the cache hits or it reaches content producer, and Data packet follows the reverse path of its Interest packet to return back to consumer finally. Besides, during Data packet transmission, routers decide whether to cache content according to cache strategy [20].

As shown in Fig. 1, the data flow of TCP/IP and NDN networks exhibit different characteristics. Specifically, in TCP/IP architecture, content fetching follows a client-server model, where client (user) connects to the server and requests the corresponding files. Then, the server continuously transmits data packets to the client through the same path, which is determined by a routing algorithm. With the TCP protocol, the client receives data in order and sends Acknowledgment (ACK) packets to assist the server in quickly identifying lost packets. Different from TCP with a fixed end-to-end connection, NDN consumers (users) send Interest packets to request content with a specific name, which is forwarded hop-by-hop until the requested content is found. Thus, in NDN, content may come from multiple resources, including the original content producer (the same as the server in TCP/IP) and intermediate routers that cache a content copy. Besides, as each Interest packet is forwarded hop-by-hop independently, the final source that responds to the Interest is jointly affected by current in-network cached content and forwarding strategies.

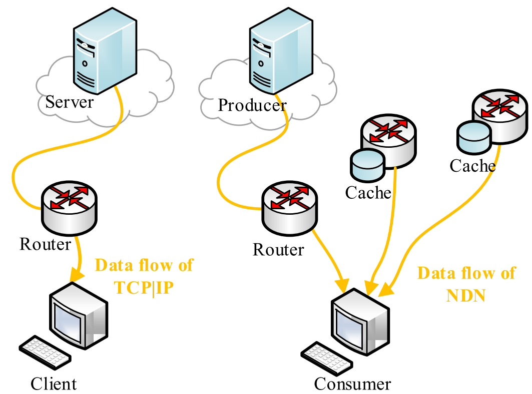


Fig. 1. Data flow features of TCP/IP and NDN networks.

Thus, the data flow of NDN is composed of multiple intermittent end-to-end transmission flows, which are transparent and uncontrollable for consumers.

III. Related Work

Until now, many researchers have tried to provide appropriate congestion control strategies for NDN networks from several aspects, including consumer-side control and hop-by-hop control [21]. We mainly consider consumer-based congestion control for its low complexity and friendly deployment. In addition, wo introduce some forwarding schemes to reveal how traffic is managed on paths.

A. Consumer-Based Congestion Control and Basic Forwarding

Consumer-based congestion control schemes adjust congestion window (or sending rate) at consumer. Most of them utilize heuristic algorithms that adjust the congestion window size (or rate) following static, pre-defined policies inherited from TCP. For example, Carofiglio et al. [12] proposed a basic congestion control algorithm following the Addictive Increase and Multiplicative Decrease (AIMD) window adjustment. It regards all sources as an entity, which decreases the total window by half when any source is congested. Liu et al. [22] utilizes deep neural networks at intermediate routers to predict data packet queue length at each interface, which is transmitted to consumers through data packets. Consumer utilizes four case predefined actions according to queue value to adjust its congestion window. There are also schemes utilizing new technologies to provide fine-grained adjustment of transmission rate. In our previous paper [13], we proposed to process data from multi-source according to their congestion degree, which helps consumers to divide data packets into "bottleneck data" and others, to treat data from multi-source differently. Then, deep reinforcement learning is utilized to predict possible source distributions and achieve fine-grained rate control.

These schemes only consider rate control and totally rely on the forwarding module [14], [15], [16] for multi-path management. These basic forwarding schemes are implemented at each router, which can obtain local information and adjust forwarding probability of each interface to distribute traffic.

For example, in [14], they try to select the interface closer to content, which is mainly designed to make full utilization of caches and reduces content retrieval delay. It does not consider different congestion degrees of available interfaces and may lack full utilization of multi-path. In [15], an Interest is forwarded in multiple disjoint paths simultaneously. By utilizing a weighted round-robin mechanism based on path delays, Interests can be distributed over multiple paths. However, transmission delay is also affected by location distribution, which may convey error congestion information. In [16], a famous Stochastic Adaptive Forwarding (SAF) method was proposed, which imitates a self-adjusting water pipe system to distribute Interests through the network. SAF can timely detect possible new sources and link failures. It also decreases the unsatisfactory interest packets passing through congested nodes. However, when consumers sense congestion, they will immediately decrease their transmission rate to avoid packet loss, which makes SAF have a slow convergence rate to transfer traffic from congested paths.

B. Joint Congestion Control and Traffic Management

Some schemes jointly consider rate control and multi-path utilization. Specifically, Schneider et al. [17] proposed an explicit congestion control algorithm based on active queue management. With this algorithm, consumer decreases congestion window according to both packet loss and congestion notification. The intermediate routers along the path also transfer traffic from the most congested interface to others when receiving marked data packets to make use of available free paths. It initially utilizes the best route (shorted path) forwarding and transfers a fixed proportion of traffic without considering congestion differences among interfaces. Ye et al. [18] proposed to determine the transmission path when sending Interests by adding PathTags. Through PathTags, this scheme forwards Interest packets to a specified path. Thus, consumers no longer treat multiple sources as one entity and maintains a rate for each path to improve control efficiency. However, in a dynamic environment, the content path may intermittently exist due to the temporary intermediate cache as content is cached or replaced. Totally determining the forwarding path may cause packet loss if a source does not exist. This often happens when the cache was replaced and may not utilize emerging sources for lacking timely detection. Carofiglio et al. [10] formulated traffic allocation for multi-path and rate control of multi-source as a Multi-Commodity Flow problem. It then solves the problem at consumer and intermediate nodes by decomposing the problem into two objectives: utility maximization of throughput (at the consumer) and network cost minimization (at intermediate nodes). It correspondingly provides optimal transmission rate and forwarding possibility. However, with time-varying sources and paths, optimal forwarding is difficult to obtain in real-time as the problem is NP-hard and time-consuming [11]. Qin et al. [23] formulates forwarding control into a local Markov Decision Process problem based on path congestion status. They utilize neuro-dynamic programming to solve this decision problem and deploy the core computing functions of the solution at edge nodes. However, it is powerless to balance the load of in-network paths, which may cause low network resource utilization.

IV. Problem Formulation and Approximate Iterative Algorithm

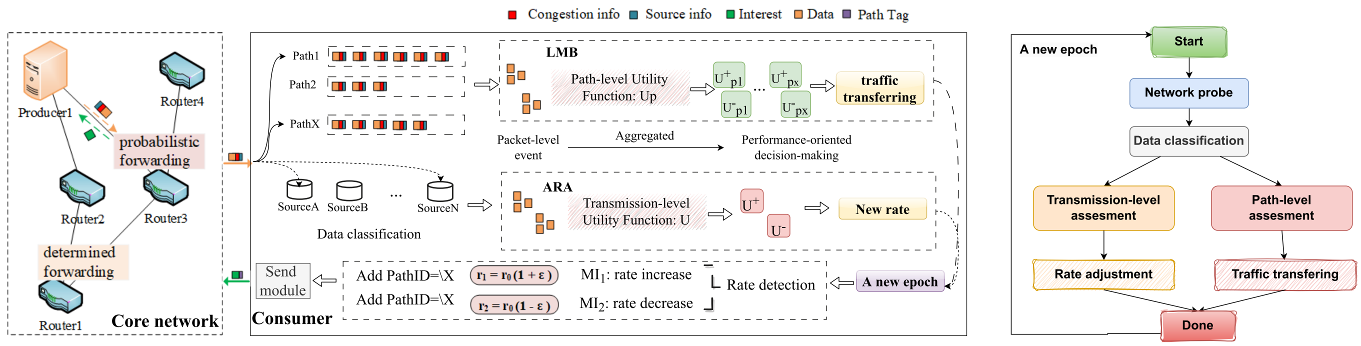
Before introducing the design details, we provide the theoretical basis for the AMM-TC scheme. Specifically, we model the resource competition among various consumers in the NDN network as a socially concave game model [24]. On this basis, we propose an adaptive online solution using online convex optimization.

A. Network Modelling

Firstly, we mathematically represent a NDN network. We suppose that the set of consumer is $C = \{c|c \in [1, N]\}$, and each consumer c can obtain content from multiple content sources through multiple paths. Then, to perform congestion control and load management simultaneously, each consumer maintains status information of two dimensions: "content-flow" and "path". Specifically, a content-flow refers to traffic from a content source (includes in-network caches and content producers) to consumer over a specific path. Data analysis over content-flow offers accurate transmission status from each source to the consumer, enabling fine-grained congestion response to heterogeneous content sources. On the other hand, a path refers to a specific consumer-producer route, which may encompass multiple content-flows along its trajectory. Besides, path serves as the operational unit for load transfer. Data analysis over path can capture congestion gaps among multiple paths, improving transmission efficiency by importing traffic into free paths. Here we suppose the set of content-flow and path for consumer c is $S_c = \{1, 2, \cdots, |S_c|\}$ and $P_c = \{1, 2, \cdots, |P_c|\}$. The content-flow set of path p is also denoted by $S_p = \{1, 2, \cdots, |S_p|\}$. Besides, we use the subscripts s, p, c, and s_p to indicate the attributes of content-flow s, path p, consumer c and content-flow s belongs to a path p.

Then, we model the behavior of multiple consumers sharing a network as an n-person game. Specifically, we denote the game by a tuple $\Gamma = (C, (X_c)_{c \in C}, (u_c)_{c \in C})$, where C is the set of consumers (players), X_c is the set of actions of consumers. We denote a specific strategy as $x_c \in X_c$, and $u_c : X_c \rightarrow \mathbb{R}$ is consumer's utility function. Besides, the joint action set of N consumers can be denoted as $Y = X_1 \times X_2 \times \cdots X_N$. We let $y = (x_1, x_2, \cdots, x_N)$ denote the strategy combination of all consumers, and y_{-c} denote the strategy combination of all consumers except consumer c. Besides, our $X_c, c \in C$ is a closed, convex and bounded action set and the utility functions u_c are concave and twice differentiable. Then, as definition in [25], the resource competing behavior of N consumers is modelled as concave games. Furthermore, through well-designed utility functions, we define it as a socially concave games [24], whose definition is shown as follows:

Definition 1: A game is socially concave if the following holds:



(a) The system architecture of AMM-CC.

(b) Control phases of the consumer.

Fig. 2. The system architecture of AMM-CC and control phases of its consumer.

- The utility of each consumer is concave in the consumer's own strategy.
- There exists a strict convex combination of the payoff functions which is a concave function. That is, there exists an n-tuple $(\lambda_c)_{c\in C}$, $\lambda_c > 0$ and $\sum_{c\in C}\lambda_c = 1$, such that $\sum_{c\in C}\lambda_c u_c(x_c)$ is a concave function in $y = (x_1, x_2, \cdots, x_N)$.
- The utility function of each consumer c, is convex in the actions of the other consumers. *i.e.*, for every $x_c \in X_c$, the function $u_c(x_c)$ is convex in $y_{-c} \in Y_{-c}$.

Moreover, the global utility is defined as the sum of consumers' utility functions. That is, $\Phi(y) = \sum_{c\in C} u_c(x_c)$, where y is a joint action, and $u_c(\cdot)$ is the utility function of each consumer c.

B. An Online Convex Algorithm

Based on former network modelling, we provide an online convex algorithm in this subsection. In general, an online algorithm has an action set $\mathcal{A}$ and $\mathcal{A} \subseteq \mathbb{R}$. At iteration t (which also means in an epoch t), the online agent chooses action $a_t \in \mathcal{A}$. Then it monitors the performance in the MI of multiple sources, using a designed cost function $f_t(a_t) \in \mathcal{F} : \mathcal{A} \to \mathbb{R}$ after committing the choice. Besides, the "regret" metrics is utilized to measure the efficiency of the online algorithm, defined as follows.

Definition 2: Given H is an online convex algorithm, and x be the optimal static solution, i.e., $x = \arg\min_{x_i}\sum_{t=1}^{T} f_t(x_i)$. In this case, the regret of the algorithm H is defined as

$$regret_H(\mathcal{T}) = \sup_{\{f_1,\cdots,f_T\}\subseteq\mathcal{F}} f_t(x_t) - min_{x\in\mathcal{A}}\sum_{t=1}^{T} f_t(x) \quad (1)$$

Therefore, to enable flexible resource utilization in NDN networks, we treat resource competition as repeated games by dividing time into consecutive Monitor Intervals (MI). Then, as illustrated in Fig. 2(b), we propose a two-layer control system to enable congestion control and load balancing across multiple paths at the consumer. Thus, at each MI, the consumer continuously interacts with the environment to probe network status. Meanwhile, it monitors performance during these micro-experiments, assessing both transmission-level and path-level performance. Based on these experimental results, each consumer then adjusts rates (ARA module) and transfers traffic (LMB module) to achieve optimal performance. Specifically, each consumer c maintains a whole sending rate r_c, which determine whole resource occupation of consumer. We define action set of online agent as $\mathcal{A} = X_c$ and $x_c = r_c$. Then, at each iteration, consumer chooses action $a_t = x_c$ and suffer a cost $f_t(a_t) = -u_c(x_c)$ after committing the choice x_c for each consumer c. Then, by using online gradient decent, consumer can update actions adaptively [26].

However, the gradient of the utility function with respect to rate in network control is difficult to obtain directly. Here we provide a sub-gradient method based on trial-error. Specifically, at each MI, consumer calculates testing rates of two possible directions based on the current rate r_c with a heuristic fixed gradient ϵ:

$$r_c^+ = r_c(1+\epsilon), \quad r_c^- = r_c(1-\epsilon). \quad (2)$$

Then, with rate detection and cost functions, consumers can estimate gradient of $f(\cdot)$ at r_c with trial-error by

$$\hat{\nabla} f(r_c) = \frac{f(r_c^+) - f(r_c^-)}{2\epsilon r_c}, \quad (3)$$

where $f(\cdot)$ can be replaced by the opposite number of the corresponding utility functions under actual application (The transmission-level utility function is U_c and path-level utility function is U_p, which will be introduced in detail later). Subsequently, we can leverage a sub-gradient descent method to minimize the cost of actions, which is equalize to maximize utility functions. For example, we update sending rate by $r_c^{new} = r_c + \eta\hat{\nabla}U_c(r_c)$, where η is rate adjustment step.

It should be noted that this strategy aims to aggressively maximize the transmission-level utility function, which directly decides resource occupation of each consumer. That is, $u_c(\cdot) = U_c(\cdot)$ and $x_c = r_c$. We will later prove that under a socially concave game, this greedy behavior also ensures the maximization of the global utility function.

V. AMM-CC: Design Details and Implementation

In this section, we elaborate the design and implementation details of the proposed Adaptive Multi-source multi-path Congestion Control (AMM-CC) scheme based on online learning. Firstly, we reveal the system architecture and provide an overview of AMM-CC to present an overall picture. Then, we introduce details of the composition modules. Finally, we reveal implementation procedure.

A. Overview

In our scheme, intermediate nodes and consumers continuously interact and cooperate through feedback information. An ARA and a LMB modules are proposed to suit flexible multi-source multi-path transmission in dynamic NDN networks. The system architecture is shown in Fig. 2(a).

Specifically, to enable multiple sources in the core network no longer transparent to a consumer, AMM-CC provides information feedback at intermediate routers to assist rate adjustment at the consumer, including source notification and congestion detection. Source notification enables each router to record its unique local identifier to form response source information. With congestion detection, consumers acknowledge congestion level by real-time feedback on the maximum queue value on the path carried in Data packets. Therefore, consumers can classify Data packets according to the source information and obtain the accurate congestion status of each content-flow.

Then, the consumer utilizes online convex optimization, with two-level meticulously designed utility functions, to guarantee smooth adjustment of the whole rate and gradually minimize congestion gaps among multiple paths, separately. As introduced in previous section, consumer explores the status of content-flows in real-time mode through micro-experiments. Then, it leverage information feedback module to classify mixed data. Subsequently, the ARA module measures and evaluates the aggregated transmission performance of all content-flows in the epoch by a transmission-level utility function, which comprehensively considers congestion status and contribution weight of each content-flow. Afterward, a gradient ascent strategy is used to realize the fine-grained rate adjustment. Furthermore, the LMB module determines the real-time congestion level to analyze path congestion status and transfer traffic to free paths. Specifically, it gets the expected rate adjustment of each path by the designed path-level utility function, obtaining the excessive traffic of the congested path and acceptable free resource on the other paths. Afterward, it adds the corresponding PathID to partial Interest packets with a stable probability, which notifies the forwarding of traffic transfer requirement. The forwarding module executes its own strategy for most of Data packets to enable exploration of new sources. It also completes hop-by-hop traffic transfers according to the instructions from consumers, improving utilization of multi-path.

We present the detailed design and implementation of the proposed AMM-CC algorithm in the following subsections. Beside, we present the mathematical notation used in system and derivation in Table.I.

TABLE I
MATHEMATICAL NOTATION USED IN SYSTEM AND DERIVATION

Parameter	**Description**
S, P, C	A set of content-flow, path, and consumers
s, p, c, s_p	The subscripts that indicates a specific content-flow, path, consumer, and content-flow on path p
X_c	A set of action for a consumer c
Y	A set of joint action of consumers
Y_{-c}	A set of joint action of consumers except consumer c
x_c	An action of consumer c
y_{-c}	Combination actions of all consumers except consumer c
$\mathcal{F}$	A convex strategy set for players
$u(\cdot)$	The utility function for players in the game
$\Phi(y)$	The global utility function for players in the game
$f(\cdot)$	The cost function for online convex algorithm
U_c	Transmission-level utility function for consumer c
$\bar{*}$	A subset of term $*$.
ϵ	The learning rate for rate detection
r, q, L, w	The request rate, queue delay, loss rate, and transmission weight metrics
α, β, γ ,φ	The hyperparameters for utility setting
U_p	The path-level utility function of path p for a consumer c (omit subscript c)
d_p	The traffic transferring quantity of path p
p_p	The traffic transferring possibility of path p

B. Information Feedback

In this part, two new fields in the data packet are designed, including "Source Info" and "Congestion Info" to transmit in-network obtained information to consumers. We also provide a "PathID" field in the Interest packet to convey traffic transfer information to the in-network forwarding module. Their functions and design details are elaborated as follows:

- Source Info: Each intermediate router adds its unique identification to this field to finally form the determined source information of the transmitted packet.
- Congestion Info: Each intermediate router modifies this field to transmit the maximal congestion degree along the path to consumers.
- PathID: Each consumer adds the identification of its intended path for traffic transfer to the Interest packets. The intermediate forwarding module will extract the information and perform traffic transfer operations.

For source notification, rather than using global unique identification of routers, each router utilizes local identifications to distinguish outgoing ports, supporting scalability. Specifically, each router uniquely identifies its own sending port and adds the corresponding ID of the port where the Data packets sent out to the "Source Info" field. As set in [27], in our design, we allocate $64bit$ to support maximum hops number of 8 with maximum port number of $2^8 - 1$. When a consumer receives Data packet, each $8bit$ data represents the local port number of a router of a new hop, finally forming a unique source identifier.

Besides, we choose the queue length of data packet along the path to notify precise congestion level. Considering transmission overhead, the packet only conveys the most congested information along its path. Specifically, each router monitors its local congestion status and compares it with the congestion level carried with the data packet. It modifies the value if it suffers heavier congestion. We provide a $8bit$ field to support the maximal queue length of 2^8, suitable for the initial buffer in ndnSIM.

The design of PathID information is the same as the obtained content-flow notification. It consists $8bit$ of data representing the local port number of each hop router along the path, which is the same set as in content-flow notification.

Besides, an additional $4bit$ is added as a hop count value, which is the index for the router to find its forwarding port and is added by one at each hop.

C. Adaptive Rate Adjustment

As shown in detail in the previous section, consumer do micro-experiment with testing rate calculated with current rate r_c by Eq.(2). Then, it achieves the desired optimal rate control with a sub-gradient method that maximize utility function of consumers. Therefore, we need to design a utility function that satisfies the socially concave game. Here we provide design details of the utility function. The proof that our design satisfies the socially concave game is placed in next section.

Supposing a consumer c receives data from multiple sources through content-flow denoted by S_c, and it obtains the transmission characteristics of each content-flow after distinguishing data packets according to the content-flow identification. To pursue higher goodput while lowering queuing delay and packet losses, we consider transmission rate, average queue length, and packet loss in our utility function. Let r_s and q_s denote the sending rates and average data packet queue length of each content-flow $s, s \in S_c$, respectively. L_s denotes packet loss. Moreover, facing mixed multi-source transmissions, AMM-CC totally includes the impact of each content-flow on the overall transmission performance. It quantifies the contribution weight and congestion impact of each content-flow. The utility function of each consumer c is designed as

$$\begin{aligned} U_c = \alpha(\sum\nolimits_{s\in S_c} r_s)^{\varphi} - (\sum\nolimits_{s\in S_c} r_s) \sum\nolimits_{s\in S_c} \{w_s(\beta \cdot q_s)\} \\ - (\sum\nolimits_{s\in S_c} r_s) \sum\nolimits_{s\in S_c} \{w_s(\gamma \cdot L_s)\}, \end{aligned} \tag{4}$$

where $\varphi \in (0, 1)$ is the parameter of power function, which is 0.9 in our design as referring to [28]. Moreover, we utilize requesting rate r_c rather than throughput for maintaining concave property of our utility function. w_s is the weight to measure the contribution of the content-flow s to the overall transmission. It is calculated by

$$w_s = \frac{G_s}{\sum_{s\in S_c} G_s}, \tag{5}$$

based on the goodput attribution of each content-flow. G_s is the estimated goodput of data received from content-flow s in the MI, and the control condition $\sum_{s\in S_c} w_s = 1$ always holds. $0 \le \alpha < 1$, $\beta > 3$, and $\gamma \ge 0$ are the parameters to decide the impact of considered characteristics, which are chosed through numerous experiments. The required characteristics of each content-flow are calculated in detail as follows:

For consumer c, who totally sends N_s Interest packets and receives M_s Data packets from content-flow $s, s \in S_c$. For each epoch, time is divided into m slots with equal length, e.g., epoch t consists of $\{t_0, \cdots, t_m\}$, and then for each source (we omit s for convenience), we can get an average queue of small time slot $\{q_t^0, \cdots, q_t^m\}$. Specifically, let $\{q_1, \ldots, q_{M_t^0}\}$ be the queue value set received at time slot t_0, and q_t^0 is

$$q_t^0 = (\sum\nolimits_{j=1}^{j=M_t^0} q_j)/M_t^0. \tag{6}$$

Then, q_s presenting the average congestion level of the content-flow s is calculated by utilizing an exponentially weighted average [29].

$$q_s = (1-g) \cdot q_t^{m-1} + g \cdot q_t^m, \tag{7}$$

iterating up to q_t^0. The most recent queue has a greater weight, which is for getting the real-time queue value in an epoch with multi-source that has different RTTs. The weight is set as 0.9, a generally utilized value.

Although the packet queue can already measure the path congestion to a certain extent, when bottleneck bandwidth drops sharply due to source/path switches, it cannot reveal overloaded capacity. Therefore, by referring [28] and [30], we utilize real-time transmission loss in each detection epoch to provide more accurate estimation of the transmission status. However, packet loss determination is a tricky task in an NDN network because there are no duplicate ACKs for fast packet loss judgment. Consumers can only detect packet loss according to Retransmission Timeout (RTO). However, it is too late for online learning to achieve timely rate adjustment in original design. Thus, we utilize the maximum RTT value of sources and Interest sending log in MI to detect packet loss. Let RTT $= (\text{rtt}_1, \cdots, \text{rtt}_{|S_c|})$ denote the vector of RTT of multiple sources. We define RTO as

$$\text{RTO} = \max\{\text{rtt}_1, \cdots, \text{rtt}_{|S_c|}\} + \Delta_{\text{rtt}}, \tag{8}$$

$$\Delta_{\text{rtt}} = \frac{\sum_{s\in S_c} \delta_s}{|S_c|}, \tag{9}$$

where δ_s is the parameter indicating RTT fluctuation and can be calculated using previous RTT samples on content-flow s. For example, δ_s can be the median deviation of the RTT samples. In our implementation, we jointly utilize Interest sending log and RTO to detect packet loss. That is, Interest packets in the sending log that are not responded within the detection cycle and last longer than RTO are regarded as lost packets. In this case, RTO can be updated continuously during the detection period, which avoids packet loss misjudgment caused by source changes. Moreover, random loss may affect estimation accuracy. To address this issue, we set specific conditions for packet loss calculation to eliminate the impact of random packet losses. We use the congestion feedback signal as an auxiliary measure, confirming congestion losses only when the packet queue exceeds the set threshold. Specifically, when the maximum packet queue value does not reach the threshold, we set packet loss as zero. Otherwise, the lost packet number is distributed to congested sources according to their composition ratios. The packet loss of the source s can be calculated by

$$L_s = \frac{N_s - M_s}{M_s} \cdot \frac{w_s}{\sum_{s\in \tilde{S_c}} w_s}, \tag{10}$$

where $\tilde{S_c}$ is the set of congested sources whose queue value exceeds the designed threshold, which is the $TH - 1$, and TH is the buffer size of intermediate routers. Thus, we can treat each source differently by the defined utility function according to its ratio and congestion status.

With the well-designed utility function, AMM-CC calculates the utility value of two testing rates, respectively. Then, it can evaluate the transmission performance of two possible directions and give a rate adjustment gradient. Basically, based on utility value, AMM-CC utilizes the estimation of gradient to pursue higher utility with an gradient ascent mathod [31]. Consequently, new rate is updated as

$$r_c^{new} = r_c + \eta(\tau)\frac{U_c(r_c^+) - U_c(r_c^-)}{2\epsilon r_c}, \tag{11}$$

where $\eta(\tau)$is a confidence amplifier adopted to increase the convergence rate. It is a monotonically non-decreasing function assigning a real value $\eta(\tau)$ to any integer $\tau \geq 0$. The amplifier increases from one to $\eta(\tau)$ if consumer makes τ consecutive decisions to change the rate in the same direction. It is set to be one if the rate adjustment direction is reversed. Besides, when online learning detects abnormal performance, such as continuous zero queue value and heavy packet loss, it will adjust the rate in one direction according to the fixed gradient. Then, it monitors the calculated utility value and network performance to decide whether to go on or return to normal rate detection. These tricks are designed further to improve the performance of online learning agents in dynamic environments.

D. Lightweight Multi-Path Balancer

The LMB module is designed to coordinate consumer rate control and forwarding modules, enabling lightweight and precise load management among multi-path in NDN networks. Here we provide a path-level utility function to measure performance for each path and transfer traffic among multiple paths adaptively.

Firstly, the path is identified by the longest content-flow identification, to which the subset content-flow identities of the intermediate cache also belongs. In each epoch and for each consumer c, we calculate the utility of the path p as (We omit subscript c of terms except P_c.)

$$\begin{aligned} U_p = \alpha(\textstyle\sum_{k\neq p\in P_c} r_k + r_p)^{\varphi} - (\textstyle\sum_{k\neq p\in P_c} r_k + r_p) \\ \cdot \left\{\textstyle\sum_{s_p\in S_p} w_{s_p}(\beta \cdot q_{s_p} + \gamma \cdot L_{s_p})\right\}, \end{aligned} \tag{12}$$

where P_c is the path set of consumer, and S_p is set of content-flow belongs to path p. Besides,

$$w_{s_p} = \frac{G_{s_p}}{\sum_{s_p\in S_p} G_{s_p}}, \tag{13}$$

denotes the weight and average queue length of content-flow s_p in the path p. r_p is the transmission rate of path p and $r_k, k \neq p \in P_c$ are rates of other paths. The average queue length of the content-flow s_p is q_{s_p}, which can be calculated by Eq.(7). L_{s_p} is the packet loss rate of content-flow s_p. It is zero if the maximum queue length is below the threshold; otherwise, we have $L_{s_p} = L_s$.

Subsequently, we utilize the gradient descent to calculate the expected rate adjustment of the path p as

$$g_p = \begin{cases} \dfrac{U_p({r_p}^+) - U_p({r_p}^-)}{{r_p}^+ - {r_p}^-} & , \quad r_p^+ \neq r_p^- \\ 0 & , \quad {r_p}^+ = {r_p}^- \end{cases} \tag{14}$$

where ${r_p}^+ = {r_c}^+ \cdot {w_p}^+$, and ${r_p}^+ = \sum_{s_p\in S_p} r_{s_p}^+$ represents the rate of the path p in the rate probing phase of r_c^+. Similarly, ${r_p}^- = {r_c}^- \cdot {w_p}^-$ and r_p^- represents the rate of the path p in the rate probing phase of r_c^-. Therefore, If $r_p^+ = r_p^-$ due to the change of the path weight, we set $g_p = 0$. Otherwise, r_p is modified in subsequent epochs.

Algorithm 1 The Process of AMM-CC for a Consumer

```
Parameters:
r_c0: the initial rate of the consumer c;
ε: gradient for rate detection;
SlowStart (r_c0);
while TRUE do
    Obtain the current rate r_c
    Prepare r_c^+, r_c^- with Eq. (2)
    Detect the network with rates of r_c^+ and r_c^-
    Collect returned data Data
    r_c^new = UpdateRc(Data)
    d_p = ObtainTS(Data,r_c^new)
    Calculate Possi={p_p|p ∈ P_c} using Eq. (18)
    Perform the traffic transfer with Possi;
end
Function SlowStart (r_c0):
    Initial the transmission rate as r_c0
    while TRUE do
        Detect the q value carried in the received data
        packet
        Obtain the current rate r_c
        if no zero queue then
            break;
        else
            r_c^new=r_c(1 + ε)
        end
    end
end
Function UpdateRc (Data):
    Obtain the content-flow information from Data
    Calculate U_p(r_p^+) and U_p(r_p^-) using Eq. (12)
    Update the r_c using Eq. (11)
    return r_c^new
end
Function ObtainTS (Data,r_c^new):
    Obtain the path information from Data
    Calculate g_p for each path using Eq. (14)
    Calculate Δ_{r_p} for each path using Eq. (16)
    Screen out the most congested path p_j
    if q_j − min{q_p, p ∈ P_c ≤ 3 then
        d_p = 0, p ∈ P_c;
    return d_p
end
```

When transferring traffic between paths, not only the expected rate change under the current congestion status needs to be considered, but also the proportion of different paths needs to be involved in the decision-making. Thus, we estimate the forwarding ratio of path p by historical data in detection phases:

$$\hat{pw}_p = \frac{\sum_{s_p \in S_p} G_{s_p}}{\sum_{p \in P} \sum_{s_p \in S_p} G_{s_p}}, \tag{15}$$

Then, we calculate the expected traffic transferring by considering the different transmission states measured by the rate changes on the path. Specifically, the traffic transfer value of each path p is obtained by

$$\Delta_{r_p} = \hat{pw}_p \cdot (g_p - \sum_{p \in P_c} \hat{pw}_p \cdot g_p). \tag{16}$$

where $\Delta_{r_p} < 0$ means the consumer needs to remove part of the traffic of path p to other paths, which is achieved by adding traffic of other paths, with $\Delta_{r_p} > 0$. Thus, the excessive traffic of the most congested path p_j, determined by the smallest Δ_{r_p}, is transferred to the other paths

$$d_p = \min\{\triangle_{r_p}, [-\triangle_{r_j}]^+\}, p \in P_c, p \neq j. \tag{17}$$

Herein $[x]^+ = x$ if $x \geq 0$ and is zero when $x < 0$, which is utilized to avoid traffic transfer when all paths are free. The minimum condition is to avoid transferred traffic exceeding the capacity of non-congested paths. We start the load transfer when the congestion difference exceeds a threshold to avoid frequent unnecessary traffic transferring. That is, if $q_j - min\{q_p, p \in P_c\} \geq 3$, the traffic shifts are applied, according to the order of $\triangle_{r_p}$ from largest to smallest. Likewise, excess traffic from other congested paths is sequentially diverted to non-congested paths. Thus, consumer determines the precise traffic amount that the forwarding module should move to the path p, which will not exceed the accommodated amount of free paths. Finally, for more smooth control, we implement the determined forwarding with a stable possibility as

$$p_p = \frac{d_p}{r_c^{new}}. \tag{18}$$

Thus, the consumer normally sends Interest packets and adds PathID in the packet with the possibility of p_p. Finally, the forwarding module obeys the instructions to apply accurate traffic diversion.

The details of the process of AMM-CC at each consumer are shown in **Algorithm.**1.

E. Implementation of AMM-CC

In this section, we give the implementation details of the algorithm. Firstly, we introduce the rate detection design to achieve efficient rate detection in the NDN network. Then, we present how consumer feedback information to the forwarding module in the network to realize load transfer requirements.

1) Efficient Performance Detection: In NDN network, data packets are from multiple sources with different transmission delay, where transmission data of different detection cycle may mixed. It may confuse online learning model and influence the behavior of rate adjustment. Besides, this natural mixture cannot be differentiated simply by dividing detection periods. In our design, we provide a detection method to equalize the impact of data mixing at different rate detection stages.

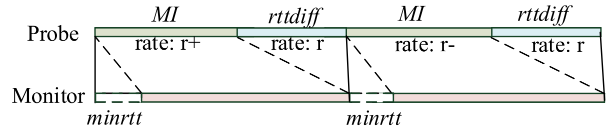


Fig. 3. Rate detection cycle and transmission monitor design of AMM-CC.

Additionally, we collect the entire transmission data corresponding to the detection rate and analyze it in detail to assess performance.

Our rate detection and transmission monitoring cycles are illustrated in Fig. 2. Specifically, we begin detection at an increased rate, identified as r^+, with a detection duration of MI. Then, detection occurs at the original rate r over a period of $rttdiff = maxrtt - minrtt$, where $maxrtt$ and $minrtt$ denote the maximum and minimum RTT values, respectively. Subsequently, detection continues at a decreased rate, identified as r^-, with a detection duration of MI, which is followed by another period of detection at the original rate r over $rttdiff$ again. Simultaneously, to ensure that consumers receive all transmission data related to speed increases and decreases, data collection begins after a period of $minrtt$ for both acceleration and deceleration detections. The collection duration is set to $MI + rttdiff - minrtt$.

Therefore, by introducing the original rate between the acceleration and deceleration detection processes, we ensure the segregation of data in these two phases. Additionally, we extend the data collection duration, allowing consumers to gather all transmission data related to both the increased and decreased rate detections. This also includes additional data at the original rate (r), which naturally intermingles with data from the acceleration and deceleration stages due to variations in RTT among multiple content sources. Besides, the mixing ratio of the original data in these two stages remains highly similar. Consequently, when evaluating the performance of the two testing rates, we effectively mitigate interference from mixed data, thereby maintaining the accuracy of the testing outcomes.

2) Multipath Traffic Transferring Realization: Multipath traffic transferring takes effect when consumer detects a gap (identified by the average queue value on the path) between the most congested path and others. Thus, it will not disturb basic forwarding functions and always tries to make full utilization of free paths and improve transmission efficiency by using multi-path effectively. Considering the possibility of multiple consumers concurrently making traffic transferring requests, the intermediate router can set a maximum threshold STH of determined forwarding in each epoch.

To realize the multi-path traffic transfer requirement conveyed by the consumer, it only needs a simple modification to the basic forwarding strategy, which satisfies basic forwarding functions. Specifically, it should check passing Interest packets to get the PathID that signals the forwarding interface at each hop. That is, if an Interest packet does not carry PathID, it performs probabilistic forwarding according to the forwarding module. Otherwise, it forwards the Interest to the specified path, where it extracts the final $4bit$ of PathID as an index i and gets its forwarding port from the ith $8bit$ in pathID.

The detailed information about PathID and packet format is illustrated in IV.B.

VI. Theoretical Analysis of AMM-CC

In this section, we provide an analysis of AMM-CC, including proof that our utility design satisfies the requirement of socially concave game. AMM-CC algorithm can converge to a Nash equlibrium under the subgradient descent algorithm. Moreover, AMM-CC algorithm can maintain stability.

A. Representation of Utility Function

Here we assume that the source composition is fixed and consumers competing for bandwidth of a bottleneck link. We first consider the scenario where all sources of consumers compete for the same transmission bottleneck. The total transmission rate of multiple content-flows of the consumer c is $r_c = \sum_{s \in S_c} r_s$. The total resource occupation in network is $R = \sum_{c \in C} r_c$. It needs to be noted that we only analyze consumers with static paths and source distribution. When the load of multiple paths is unbalanced, we assume that the queue length of other non-bottleneck content-flows is zero and congested content-flows have the same congestion degree. Specifically, we suppose the set of path goes through the competing bottleneck link l with capacity B is $\bar{P}_c$. That of the content-flow is $\bar{S}_c$. The rate of consumer c in the competing bottleneck link is $\bar{r}_c = \sum_{p \in \bar{P}_c} r_p$. We denote $\bar{R} = \sum_{c \in C} \bar{r}_c$ as the total sending rate of consumers through the bottleneck link. When intermediate routers maintain a FIFO queue and implement tail drop when congestion occurs. The change rate of queue length is $\bar{R} - B$, and the queue size will change from q_0 to $(\bar{R} - B) \cdot \tau + q_0$ after testing epoch τ. Then, the average queue length in this epoch is $(\bar{R} - B) \cdot \tau/2 + q_0$. With the drop tail strategy, the packet loss rate is $(\bar{R} - B)/B$. For each consumer, the utility function in Eq. (4) can be expressed as

$$U_c = \alpha(r_c)^{\varphi} - r_c \sum_{s \in \bar{S}_c} w_s \Big(\beta \frac{(\bar{R} - B)\tau}{2} + q_0 + \gamma[1 - \frac{B}{\bar{R}}]^{+}\Big), \tag{19}$$

where w_s is the proportion of content-flow s, $[x]^{+} = x$ if $x \geq 0$ and is zero when $x < 0$. When LMB balances load on multiple paths, all paths have the same congestion status. In this case, $\bar{P}_c = P_c$ and $\bar{R} = R$. Thus, we can abstract multiple bottlenecks with the same congestion level into a single bottleneck. The analysis in these two scenarios is consistent.

B. Socially Concave Game Guarantee

Here we prove that the utility function Eq.(19) can guarantee the socially concave game. We give one by one the proof that U_c satisfies the three conditions in Definition I.

1. We rewrite the utility function Eq.(19) as

$$U_c = \alpha(r_c)^{\varphi} - r_c \sum_{s \in \bar{S}_c} w_s \cdot \Big(\beta \frac{(r_c + r_{-c} - B)\tau}{2} + q_0 + \gamma[1 - \frac{B}{r_c + r_{-c}}]^{+}\Big). \tag{20}$$

To prove the utility of each consumer is concave in its own strategy, we calculate the second differential of r_c with respect to U_c as

$$\frac{d^2 U_c}{d^2 r_c} = \alpha \cdot \varphi \cdot (\varphi - 1) \cdot (r_c)^{\varphi - 2} - \sum_{s \in \bar{S}_c} w_s \beta \tau - \sum_{s \in \bar{S}_c} w_s \gamma \frac{B}{(r_c + r_{-c})^2} - r_c \sum_{s \in \bar{S}_c} w_s \gamma \frac{2B}{(r_c + r_{-c})^3}, \tag{21}$$

where $\varphi - 1 < 0$ and all parameters are positive, thus $\frac{d^2 U_c}{d^2 r_c} < 0$. Therefore, U_c is concave for r_c, and the first condition satisfies.

2. The sum of utility can be written as $\Phi(y) = \sum_{c \in C} U_c(r_c)$, $y = (r_1, r_2, \cdots, r_n)$. Then, as R is a linear function of y and $\bar{R}$, we only need to prove that $\Phi(y)$ is concave for $\bar{R}$ to equally prove that $\Phi(y)$ is concave for y. Here we calculate the second derivative to $\Phi(y)$ with respect to $\bar{R}$, and we have

$$\frac{d^2 \Phi(y)}{\bar{R}^2} = \frac{d^2 \sum_{c \in C} r_c^{\varphi}}{d\bar{R}^2} - \frac{2 \sum_{s \in \bar{S}_c} w_s \gamma}{B}, \tag{22}$$

where r_c^{φ} is a concave function when $\varphi \in (0, 1)$, and then $\sum_{c \in C} r_c^{\varphi}$ is concave. Therefore, the first part of the function is concave. Besides, as $-2 \sum_{s_p, p \in \bar{P}_c} w_{s_p} \gamma / B < 0$, the second term in the utility function is also concave. Therefore, the sum of utility is strictly concave in y. Thus, the second condition satisfies.

3. We denote a joint action of other consumers as y_{-c}, and the sum rate of other consumers as $r_{-c} = \sum_{i \neq c, i \in C} r_i$. As r_{-c} is a linear function of $r_i, i \neq c, i \in C$, we only need to prove that U_c is convex in r_i to prove that U_c is convex in other player's strategy. Here we calculate the second derivative to U_c with respect to $r_i, i \neq c$, and we have

$$\frac{d^2 U_c}{d^2 r_i} = 2 r_c \sum_{s \in \bar{S}_c} w_s \gamma \frac{B}{(r_c + r_{-c})^3}, \tag{23}$$

where all of the parameter are positive. Therefore, $\frac{d^2 U_c}{d^2 r_i} > 0$ and the third condition satisfies.

To sum up, consumers' competition for resources on paths satisfies the socially concave game. □

C. Convergence Analysis

In this part, we analyze convergence performance of AMM-CC and we provide the following Lemma.

Lemma 1: AMM-CC enables convergence to a rate $r^{\star'}$ that is no more than $2\epsilon r^{\star'}$ away from the optimal rate $r^{\star}$.

Proof: To demonstrate the convergence of the algorithm, we analyze the behavior of the rate updates and finally show how the gap between successive rates diminishes until convergence.

Firstly, we define $\delta r_t = r_{t+1} - r_t$. According to the rate update formula, we have $r_{t+1} = r_t + \eta \hat{\nabla} U_c(r_t)$ and thus $\delta r_t = \eta \hat{\nabla} U_c(r_t)$. Then, to prove the convergence of algorithm, it is equivalent to proving that $\delta r_{t+1} - \delta r_t \leq 0$ and $|r^{\star'} - r^{\star}| \leq 2\epsilon r^{\star}$, where $r^{\star'}$ is the convergence point of the algorithm and $r^{\star}$ is the optimal rate.

Specifically, since the utility function U_c is a concave function, we have $\frac{d^2U_c}{d^2r_c} < 0$. Thus, the slope of the secant line formed by two points $(r_1, U_c(r_1))$ and $(r_2, U_c(r_2))$, where $r_1 < r_2$, will decrease as r_1 increase. Then, suppose that r_1 corresponds to $r^-_{c,t}$ and r_2 corresponds to $r^+_{c,t}$. The sub-gradient $\nabla U_c(r_c)$ based on trial-error is also decrease as r_c increase. Then, we have $\delta r_{c,t+1} - \delta r_{c,t} = \eta(\hat{\nabla} U_c(r_{c,t}) - \hat{\nabla} U_c(r_{c,t-1})) \leq 0$. Therefore, the algorithm is gradually reducing the gap between $r_{c,t+1}$ and $r_{c,t}$, which indicates that sending rate r_c is gradually converging. Finally, when $\hat{\nabla} U_c(r_{c,t}) = 0$, we achieve $r_{c,t+1} = r_{c,t}$, at which the algorithm reaches its convergence point $r^{\star'}$ and the rate does not change again until network changes again.

Moreover, as the slope of the secant line formed by two points may achieve zero when $U_c(r^+_{c,t}) = U_c(r^-_{c,t})$. In this case, AMM-CC converge to a local optimal point for using experience-gradient, and $r^{\star'} - r^\star \leq 2\epsilon r^{\star'}$ as there must be $r^{\star'}(1+\epsilon) > r^\star$ and $r^{\star'}(1-\epsilon) < r^\star$. Otherwise, AMM-CC converges to the optimal rate $r^\star$. □

Besides, we present a lemma to reveal that the role of each consumer' selfish behavior also optimizes overall transmission efficiency.

Lemma 2: Given that consumers obey n-person socially concave games and leverage gradient ascent method to maximize their utility functions aggressively. Then, the global utility defined as the sum of player's utility is also maximized.

Proof: Firstly, as proved in previous subsection, in a socially concave game, our global utility function $\Phi(y)$ is a concave function with respect to the combination of actions y. Additionally, consumers' rates (strategies) can take any value within the interval $[0, B]$, which forms a convex set. Then, according to the properties of concave functions, the local maximum point of a concave function over a convex set is also the global maximum point. Subsequently, we only need to prove that each consumer's strategy make $\Phi(r_c, r_{-c})$ achieves its local maximum point where $\frac{\partial \Phi(r_c, r_{-c})}{\partial r_c} = 0$ for all $c \in C$. Specifically, each consumer c achieves optimal strategy $r^\star_c$ at a point where $\frac{dU_c}{dr_c} = 0$. As each consumer adjusts sending rate independently, $r_i, i \neq c, i \in C$ is independent variable to r_c. Thus, we have $\frac{\partial \Phi(r_c, r_{-c})}{\partial r_c} = \frac{dU_c}{dr_c}$. Therefore, when consumer c achieves optimal rate $r^\star_c$, the global utility function also reaches a point $\frac{\partial \Phi(r^\star_c, r_{-c})}{\partial r^\star_c} = 0$. Then, when all consumers achieves their optimal rate, the global utility function also reaches its local maximum point. As its local maximum is also the global maximum, the global utility function is also maximized. □

D. Stability Analysis

In this part, we provide that through the utility-based rate control and proposed lightweight multi-path balancer, consumer can finally achieve stability states in a specific network with a relatively stable link state and cache response condition.

Lemma 3: Given each consumer updates transmission rate using the utility function defined in Eq. (4), and uses the path utility function defined in Eq.(12) to manage multi-path, from some moment onwards, every path in the network experiences the same congestion level in any equilibrium.

Proof: Supposing a consumer has two paths, there is a congested path j with weight of w_j, and another path k has a lower congestion level where

$$\Delta_{r_k} = \hat{pw}_k \cdot (g_k - \sum\nolimits_{p \in P} \hat{pw}_p \cdot g_p) > 0, \tag{24}$$

and the traffic of path j is transferred to path k. In this case, no matter how consumer adjusts the whole rate, in the static forwarding possibility on the two paths, the congestion differences of the path j and path k will decrease. Thus, by using Eq.(12), consumers can timely achieve the congestion gap among its multiple paths, and smoothly transfer traffic to enable every path to experience the same congestion level.

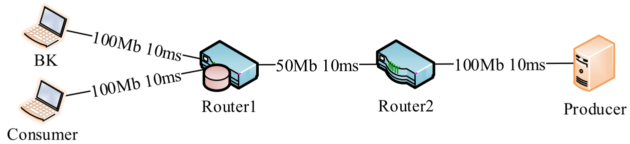
Then, we reveal that consumer updates its rate to enable all paths in the network to experience the same congestion level. We analyze how it behaves under two scenarios of the link being under-utilized and overused. When link is under-utilized, consumers see lower congestion information, including near-zero queue length and no packet loss. Then, we have $\bar{R} < B$ and $\sum_{c \in C} \frac{dU_c}{dr_c} > 0$. In this case, consumers will increase their rate in the following epoch until network resources are fully utilized. When the congestion occurs, that is $\bar{R} > B$, the average queue length grows, and even packet loss occurs. When consumers continue to grow their rates, they have $\sum_{c \in C} \frac{dU_c}{dr_c} < 0$. Then, they obtain a negative gradient, and the rate decreases. With completion, finally each consumer have the same total sending rate. Moreover, since $\frac{dU_c}{dr_c}$ is bounded, AMM-CC updates rate stably at each time step until reach the convergence rate $r^{\star'}$ (proved in previous subsection). □

VII. Performance Evaluation

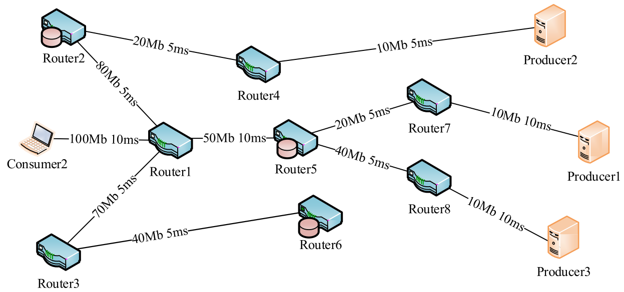
In this section, we evaluate the performance of the proposed AMM-CC via a simulation environment based on an ns-3-based NDN simulator, ndnSIM [32] (version 2.8) under the Ubuntu 18.04 virtual machine with 4cores and 8GB memory. Our comparison schemes include some latest congestion control schemes in NDN networks. Specifically, we compare AMM-CC with state-of-the-art congestion control solutions, including Practical Congestion Control (PCON) [17], Delay-based Path-specified Congestion Control (DPCCP) [18], Optimal Multipath Congestion Control and Request Forwarding (OMCCRF) [10], and ICP [12]. Except for PCON that we directly used its published source code [33], we rewrite and implement the outer algorithms by ourselves and utilize parameter settings that are consistent with the description in the papers. We measure the transmission efficiency and reliability of algorithms through goodput, average queuing delay, and packet loss. Next, we introduce test scenarios and corresponding results.

A. Single-Path Multi-Sources Environment

We first evaluate the performance of AMM-CC in a single path with multiple sources, including an intermediate router with caches and content producers. Specifically, as shown in Fig. 4(a), Consumer denotes the consumer that sends Interests, Producer denotes content producers publishing and storing contents, and the Router connecting consumers and

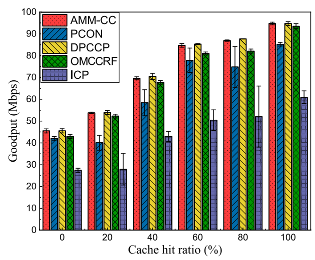


(a) The linear topology.

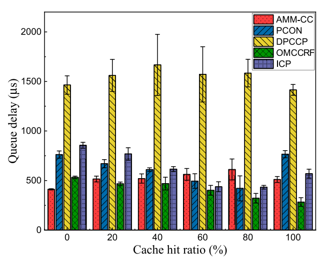


(b) The multi-path topology.

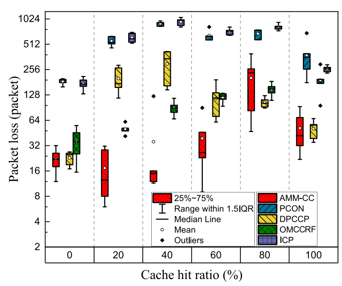
Fig. 4. The details of topologies.



(a) Performance of goodput.



(b) Performance of packet loss.



(c) Performance of queuing delay.

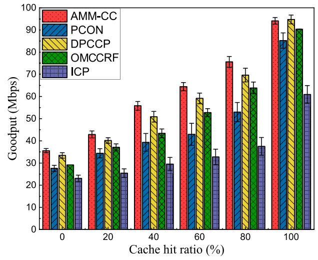
Fig. 5. Transmission performance in a stable environment.

producers holds a cache that can cache content for quick responses. BK is a node that sends Interest requests randomly to manufacture background flows, which simulates the slight network state fluctuations that users may experience during transmission. In the implementation, the bandwidth occupied by the background flow varies within the range of 0 to 5 Mbps. Specifically, BK nodes select a random number $n_{bk} \in (0, 1)$ at every MI (online learning update time) and then send Interest requests at a rate of $r_{bk} = 500 \cdot n_{bk}$ packets per second. We test different cache settings from no-cache response to whole cache response, with various link settings including state link information, varying RTTs and varying link bandwidth. Besides, we utilize the default routing strategy, and run every test twenty times to summarize the average results.

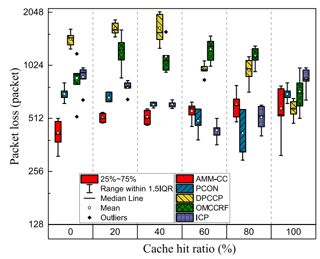
Stable scenario: In Fig. 5, we present the transmission performance of different algorithms in a stable environment. With various cache hit ratios from 0% to 100%, the goodput of these algorithms shows a constant increase. It shows that after the well-designed control, utilizing an intermediate cache can significantly improve transmission efficiency. The goodput can increase by up to 89.0% with 80% of cached responses compared to content retrieval only from the producer. In various cache response scenarios, AMM-CC is more stable and can achieve higher goodput. Specifically, when there is no cache response, it can improve the goodput by 8.33%, 5.81%, and 65.56% compared to PCON, OMCCRF, and ICP, respectively. When intermediate routers can respond to 80% requests, the imporvement of goodput are 16.18%, 5.97%, and 67.11% compared with the three others, respectively. DPCCP with several well designed control routines performs consistently well as AMM-CC in terms of goodput. However, in all considered cache response scenarios, AMM-CC can achieve much lower queuing delay and packet loss compared with DPCCP and the other algorithms. In most scenarios, it can limit the number of lost packets to 100 packets and keep the queuing delay within 600 microseconds. Specifically, taking the 40% cache response scenario as an example, AMM-CC can reduce packet loss number compared to others, and reduce queuing delay by $90.18\ \mu s$ to $1146.80\ \mu s$. The queuing delay of DPCCP is higher because it sets a high static threshold to detect congestion. When data from multiple sources are mixed, it obtains a much lower congestion degree of the most congested path due to congestion degree dilution by data from other non-congested sources. Then, it continues to increase sending rate resulting in higher queuing delay and packet loss. AMM-CC adopts a more flexible rate adjustment strategy based on the utility function, thus, maintaining higher transmission efficiency.

Bandwidth varying scenario: In the bandwidth-varying scenario, twenty timing points are selected in 200s, and then the bandwidth of bottleneck link $Router1$ to $Router2$ is randomly changed from $10Mbps$ to $80Mbps$. The evaluation is to show the performance of algorithms in dynamic NDN networks with unexpected bandwidth changes.

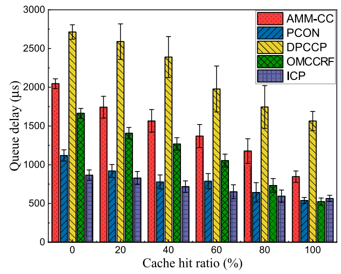
As shown in Fig. 6, goodput also increases as cache hit ratio grows. Basides, AMM-CC also outperforms the state-of-the-art algorithms in this scenario. That is, AMM-CC achieves the best goodput under various cache hit ratios, with packet loss and queue delay in normal ranges. Specifically, it can improve the goodput by 50.12%, 22.16%, 97.24%, and 9.13% compared to PCON, OMCCRF, ICP, and DPCCP at 60% cache hit ratio, respectively. In the bandwidth-varying scenario, AMM-CC utilizes the utility function to obtain a more flexible adjustment of rate, and thus, it shows better adaptation to changing bandwidth. Queue delay and packet loss increase compared with that in stable scenarios, which is because congestion usually occurs with unexpected bandwidth decreasing. In this case, AMM-CC can limit packet loss with 600 packets and queue delay below $2000\mu s$. Specifically, compared with DPCCP, in the 40% cache response scenario, AMM-CC decreases the packet loss and queue delay by $3.1\times$ and 34.49%, respectively. Besides, when the cache hit ratio is 100%, data packets return via a more stable path with higher bandwidth, improving the consumer's throughput. Therefore, algorithms perform significantly better with a cache hit ratio



(a) Performance of goodput.

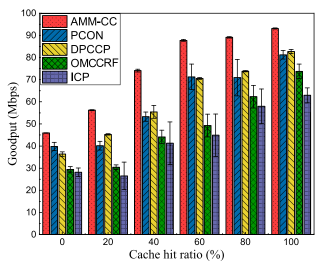


(b) Performance of packet loss.

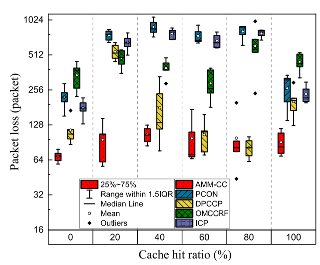


(c) Performance in queuing delay.

Fig. 6. Transmission performance in bandwidth varying environment.



(a) Performance of goodput.



(b) Performance of packet loss.

Fig. 7. Transmission performance in rtt varying environment.

of 100% compared to other cases. Moreover, queue delay of all algorithms decreases as cache hit ratio increases, is because data packet travelling the varying bandwidth link decrease.

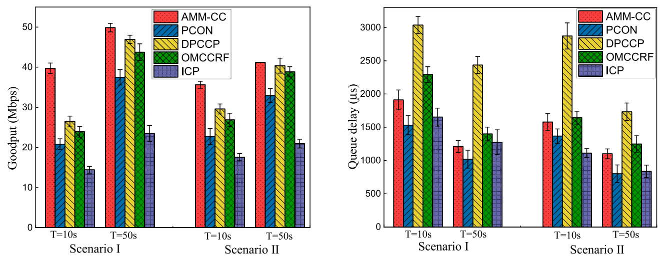
Varying delay scenario: Then, we evaluate how algorithms perform when RTTs are varying, where we add random changeable RTTs from $10ms - 30ms$ to the link from $Consumer$ to $Router1$. The time interval between two cases of changes can be as small as $0.001s$ and variation as little as $1ms$. In Fig. 7, we present the performance of compared algorithms. Compared with the performance in stable scenario, OMCCRF and DPCCP perform less better in terms of all evaluation metrics. It is because when RTT varying are applied in NDN networks, DPCCP and OMCCRF adjust rate based on the minimum RTT value of the path and may make wrong decisions. They may mistakenly regard normal delay increase as congestion, resulting in performance decay. Specifically, the goodput gap between DPCCP and AMM-CC is significantly widened, compared to the above scenarios. And again, taking the 40% cache response scenario as an example, AMM-CC increases the goodput by 33.57% and 68.03% compared with DPCCP and OMCCRF. respectively. It needs to be noted that though AMM-CC also rely on RTTs to update its detection cycle and packet loss, it updates RTT values during detection and does not rely on minRTT. In addition, PCON plays well in this scenario for relying on explicit congestion marks.

B. Multi-Paths Multi-Sources Environment

In this subsection, we introduce how AMM-CC performs when there are multiple paths with multiple content response sources. Specifically, as shown in Fig. 4(b), there are multiple paths that separately connect the consumer to three content providers; along the path, there are intermediate routers with caches, which can provide a faster content response. Besides, we evaluate scenarios where various sources are constantly being activated and respond to consumers alternatively, which is common in dynamic environments. Moreover, we consider scenarios where multiple sources transmit data simultaneously. It needs to be noted, facing multiple paths, DPCCP maintains multiple windows for multiple paths. PCON maintains a whole window for multiple paths and transmits based on the best route. It constantly transfers a fixed percentage of traffic from congested paths to other paths in intermediate router when receiving a congestion mark. OMCCRF utilizes its proposed forwarding strategy to distribute traffic based on PIT length. ICP only maintains a whole window for multiple paths. AMM-CC utilizes LMB module to transfer traffic from the congested path to other paths by consumer-side analysis and adaptively adds PathID in interest packets to specify the forwarding path of transferring traffic. Besides, as we have comparatively tested and presented the performance of different cache ratios at routers, in this scenario, we only present the performance of algorithms in varying cache ratios with 0 or 100%.

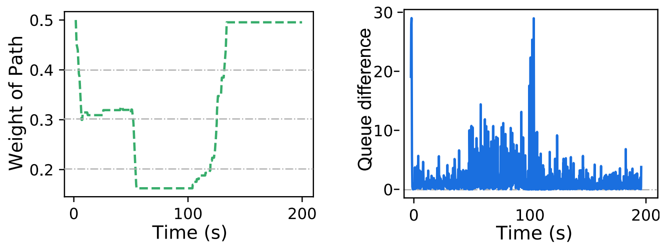
Source switching scenario: Firstly, we present the source-switching scenario, where Interest packet are constantly responded to by changing sources. Specifically, there are six content sources considered, including three producers ($Producer1 - Producer3$) and three caches ($Router2$, $Router5$, and $Router6$). They are activated alternatively to respond to consumer requests. To show the influences of different characteristics of sources, we examine two scenarios to verify the excellent performance of AMM-CC. Specifically, in scenario 1, two data sources with the largest gap in transmission delay and bandwidth, $Rrouter2$ and $Producer1$, are exchanged to answer the Interest packet. In scenario 2, the whole six sources answer the interest request in a specific order from producer to routers along the path, repeatedly. For each scenario, we examine two switching intervals, including per ten seconds and per fifty seconds.

The results are shown in Fig. 8. We can observe that AMM-CC also outperforms the state-of-the-art algorithms in the two scenarios. Specifically, when the switching interval is $10s$, AMM-CC improves goodput by 87.5%, 50.37%, 66.11%, 65.98%, and 175.69% in scenario1, and 56.62%, 20.42%, 32.61%, and 102.62% in Scenario 2, compared with PCON, DPCCP, OMCCRF, and ICP, respectively. In scenario 1, the great gap in bandwidth and delay makes the algorithms with slow convergence rate, consuming a long time to probe the bandwidth, resulting in lower goodput than in Scenario 2. When testing duration is 10s, OMCCRF with a slow converge speed causes bandwidth underutilization at most of the time. DPCCP with more adaptive rate adjustment design can converge faster, resulting in better utilization of bandwidth and higher throughput. In these two scenarios, AMM-CC, which



(a) Performance in goodput.
(b) Performance in queuing delay.

Fig. 8. Transmission performance in multi-source changing environment.



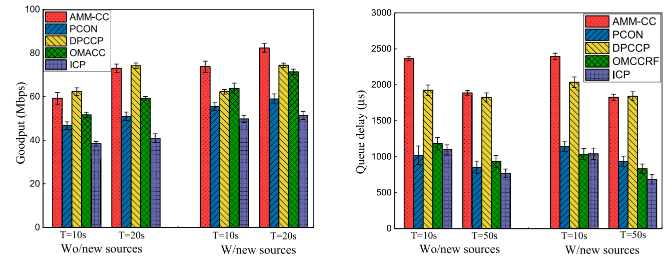
(a) Weight changes of Path1.
(b) Queue difference of paths.

Fig. 9. Traffic transfer performance of proposed LMB.

adjusts its transmission rate based on utility function, can quickly adapt to varying sources. It can continuously increase the transmission rate when it detects that the network is idle, having a faster convergence rate, and decreases the rate when heavy congestion occurs. Moreover, when the switching interval is fifty seconds, all algorithms have higher goodput than that of switching per ten seconds. It is because, in the former, algorithms have more time in a steady state to transmit more data. Besides, AMM-CC has a lower queue delay compared with DPCCP as AMM-CC can detect congestion with queue length fed back by intermediate routers and adjusts the rate before congestion occurs.

Simultaneous varying multi-source transmission scenario: In this scenario, we illustrate how AMM-CC performs when multiple sources in multiple paths transmit data simultaneously. In our evaluation, we also consider two situations, including $Wo/new\quad sources$ and $W/new\quad sources$ scenarios. In the first scenario, we show the influence of stable existence of multiple paths and caches. Specifically, consumer requests content by sending Interest packets, which are forwarded by routers to three producers. Thus, at the beginning of transmission, consumers can receive content from three producers through three paths simultaneously. After $50s$ of transmissions, $Rrouter5$ with cached content starts to answer requests, and the $Router2$ starts to answer requests after $100s$ of transmission. In the second scenario, a new content source appears during transmission. Specifically, during transmission, the $Router6$ with content cache also joins the transmission and provides content after 5s of transmission.

Before showing detailed performance, we firstly provide transmission weight changes of path1 (the path from $Consumer$ to $Producer1$) during transmission. The forwarding ratios to three producers are 0.5, 0.25 and 0.25, respectively. As shown in Fig. 9, at the beginning, path1 ($Consumer$ to $Producer1$) with a bottleneck bandwidth of



(a) Performance in goodput.
(b) Performance in queuing delay.

Fig. 10. Transmission performance in multi-path multi-source environment.

$10Mbps$ and forwarding ratio of 0.5 is the most congested path during transmission that block efficient utilization of other paths. With consumer-assisted forwarding, AMM-CC analyzes transmission and constantly quickly decreases the weight of path1 to 0.33 to obtain a similar congestion level to the other two paths. Then, when $Router2$ starts to respond to requests at $50s$, weights continue to decrease, allowing the consumer to take advantage of other paths. The same timely reaction occurs when $Router5$ starts to respond to interest requests resulting in increased weight on path1. The difference of data packet queue length along the most congested path and other paths also decreases as traffic is transferred from the congested path.

The detailed performance is shown in Fig. 10. In the $Wo/new\quad sources$ scenario, AMM-CC that maintains only a whole rate for a consumer can achieve approximately as high a goodput as DPCCP, showing great performance of AMM-CC in multi-path rate control. Specifically, compared with PCON, OMCCRF, and ICP, which also maintains a whole rate for consumer, AMM-CC increases goodput by 39.2% 20.75%, and 128.57%, respectively. In the $W/new\quad sources$ scenario, $Router4$ with cached content unexpectedly joins the transmission. DPCCP, which determines the transmission path in advance, is unable to find and utilize the new source, while the other three algorithms, without limitation on forwarding paths, are able to utilize new sources timely. In this scenario, AMM-CC even achieves higher goodput than DPCCP, which maintains multiple independent rates for multiple paths. Specifically, AMM-CC improves goodput by 26.76% compared with DPCCP. The results show that, traffic transferring has a great impact on the effective utilization of multi-path.

Besides, in both scenarios, AMM-CC exhibits a slightly higher queue delay due to two main factors. Firstly, compared with algorithms such as ICP, PCON, and OMCCRF that significantly underutilize bandwidth, having no data packet in queue. AMM-CC can adaptively adjust the sending rate and maintain the data packet queue within a specified threshold range. Thus, AMM-CC's higher queue delay comes from its higher resource utilization. Secondly, compared to DPCCP, the higher delay in AMM-CC arises from its higher convergence threshold setting. Despite this, the delay performance of AMM-CC across all scenarios shows that its queue delay remains within a reasonable range.

VIII. Conclusion

In this paper, we proposed an adaptive multi-source multi-path congestion control scheme, AMM-CC, for dynamic

NDN networks based on online learning. AMM-CC affords a dual-layered control mechanism to enable smooth adjustment of overall transmission rate and timely traffic transferring among multiple paths. Specifically, by a meticulously designed transmission-level utility function that jointly considers the congestion status and weight of each source, AMM-CC aggregates the contribution of the individual source to overall performance. It obtains fine-grained rate adjustment by a gradient descent method. Moreover, AMM-CC also enables consumers to monitor and calculate congestion gaps among multiple paths by a path-level utility. Thus, consumers can coordinate with the basic forwarding strategy to support precise traffic transfers and make full use of multiple paths. Through comparative evaluations with ndnSIM, we verified that AMM-CC could make full use of multiple sources and multiple paths, enabling flexible content acquisition in dynamic NDN networks compared with the state-of-the-art schemes.

REFERENCES

[1] *Cisco Annual Internet Report (2018–2023)*. Accessed: Jun. 2024. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[2] *Mobile Data Traffic Outlook*. Accessed: Jun. 2024. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast

[3] G. Xylomenos et al., "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2013.

[4] L. Zhang et al., "Named data networking (NDN) project," Univ. California, Los Angeles, Palo Alto Res. Center (PARC), Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010. Accessed: Jun. 2024. [Online]. Available: https://named-data.net/wp-content/uploads/TR001ndn-proj.pdf

[5] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity in the internet," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1260–1271, Dec. 2006.

[6] W. Wei, K. Xue, J. Han, D. S. L. Wei, and P. Hong, "Shared bottleneck-based congestion control and packet scheduling for multipath TCP," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 653–666, Apr. 2020.

[7] J. Han et al., "FMPTCP: Achieving high bandwidth utilization and low latency in data center networks," *IEEE Trans. Commun.*, vol. 72, no. 1, pp. 317–333, Jan. 2024.

[8] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, and D. S. L. Wei, "An online learning assisted packet scheduler for MPTCP in mobile networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2297–2312, Oct. 2023.

[9] R. Zhuang et al., "Achieving flexible and lightweight multipath congestion control through online learning," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 1, pp. 46–59, Mar. 2023.

[10] 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.

[11] A. Hall, S. Hippler, and M. Skutella, "Multicommodity flows over time: Efficient algorithms and complexity," *Theor. Comput. Sci.*, vol. 379, no. 3, pp. 387–404, Jun. 2007.

[12] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: Design and evaluation of an interest control protocol for content-centric networking," in *Proc. IEEE INFOCOM Workshops*, Mar. 2012, pp. 304–309.

[13] J. Yang et al., "IEACC: An intelligent edge-aided congestion control scheme for named data networking with deep reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4932–4947, Dec. 2022.

[14] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *Proc. 1st ACM Conf. Inf.-Centric Netw.*, Sep. 2014, pp. 127–136.

[15] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An on-demand multi-path interest forwarding strategy for content retrievals in CCN," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–6.

[16] D. Posch, B. Rainer, and H. Hellwagner, "SAF: Stochastic adaptive forwarding in named data networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1089–1102, Apr. 2017.

[17] K. Schneider, C. Yi, B. Zhang, and L. Zhang, "A practical congestion control scheme for named data networking," in *Proc. 3rd ACM Conf. Inf.-Centric Netw.*, 2016, pp. 21–30.

[18] Y. Ye, B. Lee, R. Flynn, J. Xu, G. Fang, and Y. Qiao, "Delay-based network utility maximization modelling for congestion control in named data networking," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2184–2197, Oct. 2021.

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

[20] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[21] 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.

[22] T. Liu, M. Zhang, J. Zhu, R. Zheng, R. Liu, and Q. Wu, "ACCP: Adaptive congestion control protocol in named data networking based on deep learning," *Neural Comput. Appl.*, vol. 31, no. 9, pp. 4675–4683, Sep. 2019.

[23] J. Qin, Y. Xing, W. Wei, and K. Xue, "Edge computing aided congestion control using neuro-dynamic programming in NDN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[24] E. Even-Dar, Y. Mansour, and U. Nadav, "On the convergence of regret minimization dynamics in concave games," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 523–532.

[25] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, p. 520, Jul. 1965.

[26] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 928–936.

[27] D. Li and J. Wu, "On the design and analysis of data center network architectures for interconnecting dual-port servers," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2014, pp. 1851–1859.

[28] M. Don et al., "PCC Vivace: Online-learning congestion control," in *Proc. 15th USENIX Symp. Netw. Syst. Design Implement.* Berkeley, CA, USA: USENIX, 2018, pp. 343–356.

[29] J. S. Hunter, "The exponentially weighted moving average," *J. Quality Technol.*, vol. 18, no. 4, pp. 203–210, 1986.

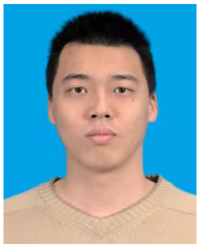
[30] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*. Berkeley, CA, USA: USENIX, 2015, pp. 395–408.

[31] S. Singh, M. J. Kearns, and Y. Mansour, "Nash convergence of gradient dynamics in general-sum games," in *Proc. 16th Conf. Uncertainty Artif. Intell. (UAI)*, 2000, pp. 541–548.

[32] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," Univ. California Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0028, 0028. Accessed: Jan. 2024. [Online]. Available: https://named-data.net/wp-content/uploads/2013/07/ndn-0028-1-ndnsim-v2.pdf

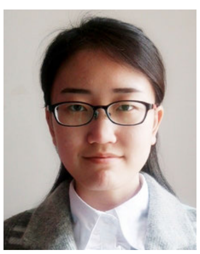
[33] (2024). *PCON Source Code*. Accessed: Jun. 2024. [Online]. Available: https://github.com/schneiderklaus/ndnSIM

Jiayu Yang (Member, IEEE) received the bachelor's and Ph.D. degrees from the School of Cyber Security, University of Science and Technology of China (USTC), in 2019 and 2024, respectively. Her research interests include future internet architecture design, transmission optimization, and network security.



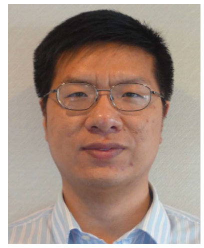
Yuxin Chen (Graduate Student Member, IEEE) received the bachelor's degree in information security from the School of the Gifted Young, University of Science and Technology of China (USTC), in July 2021. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Technology, USTC. His research interests include future internet architecture design, transmission optimization, and network security.

Kaiping Xue (Senior Member, IEEE) received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida. He is currently a Professor with the School of Cyber Science and Technology, USTC. He is also the Director of the Network and Information Center, USTC. His research interests include next-generation internet architecture design, transmission optimization, and network security. He is a fellow of IET. He serves on the editorial board for several journals, including IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Wireless Communications, and IEEE Transactions on Network and Service Management.



Jiangping Han (Member, IEEE) received the bachelor's and Ph.D. degrees from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2016 and 2021, respectively. From November 2019 to October 2021, she was a Visiting Scholar with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. She was a Post-Doctoral Researcher with the School of Cyber Science and Technology, USTC. She is currently an Associate Researcher with the School of Cyber Science and Technology, USTC. Her research interests include future internet architecture design and transmission optimization.

Jian Li (Senior Member, IEEE) received the bachelor's degree from the Department of Electronics and Information Engineering, Anhui University, in 2015, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2020. From November 2019 to November 2020, he was a Visiting Scholar with the Department of Electronic and Computer Engineering, University of Florida. From December 2020 to December 2022, he was a Post-Doctoral Researcher with the School of Cyber Science and Technology, USTC. He is currently an Associate Researcher with the School of Cyber Science and Technology, USTC. His research interests include quantum networking, wireless networks, and next-generation internet architecture. He serves as an Editor for *China Communications*.



Ruidong Li (Senior Member, IEEE) received the bachelor's degree in engineering from Zhejiang University, China, in 2001, and the Ph.D. degree in engineering from the University of Tsukuba in 2008. He is currently an Associate Professor with the College of Science and Engineering, Kanazawa University, Japan. Before joining Kanazawa University, he was a Senior Researcher with the Network System Research Institute, National Institute of Information and Communications Technology (NICT). His current research interests include future networks, big data networking, blockchain, information-centric networks, the Internet of Things, network security, wireless networks, and quantum internet. He is a member of IEICE. He is also the Founder and the Chair of the IEEE SIG on big data intelligent networking and the IEEE SIG on intelligent internet edge. He is also the Secretary of the IEEE Internet Technical Committee. He serves as the chair for some reputed conferences and workshops, and organizes the special issues for the leading magazines and journals.

Qibin Sun (Fellow, IEEE) received the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1997. He is currently a Professor with the School of Cyber Science and Technology, USTC. He has published more than 120 papers in international journals and conferences. His research interests include multimedia security, and network intelligence and security.

Jun Lu received the bachelor's degree from Southeast University in 1985 and the master's degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1988. He is currently a Professor with the School of Cyber Science and Technology, USTC. He is also the President of Jiaxing University. His research interests include theoretical research and system development in the field of integrated electronic information systems, and network and information security. He is an Academician of the Chinese Academy of Engineering (CAE).