This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

QSCCP: A QoS-Aware Congestion Control Protocol for Information-Centric Networking

He Bai, Hui Li, *Senior Member, IEEE*, Jianming Que, Abla Smahi, Minglong Zhang,
Peter Han Joo Chong, *Senior Member, IEEE*, Shuo-Yen Robert Li, *Life Fellow, IEEE*, Xiyu Wang, and Ping Lu

***Abstract*—Information-Centric Networking (ICN) is a promising future network architecture that shifts the host-based network paradigm to a content-oriented one. Over the past decade, numerous ICN congestion control (CC) schemes have been proposed, tailored to address congestion issues based on ICN's transmission characteristics. However, several key challenges still need to be addressed. One critical issue is that most existing CC studies for ICN do not consider the diverse Quality of Service (QoS) requirements of modern network applications. This limitation hinders their applicability across various applications with different network performance preferences. Another ongoing challenge lies in improving transmission performance, particularly considering how to appropriately coordinate congestion control participants to enhance content retrieval efficiency and ensure reasonable resource allocation, especially in multipath scenarios. To tackle these challenges, we propose QSCCP, a QoS-aware congestion control protocol built upon NDN (Named Data Networking), a well-known ICN architecture. In QSCCP, diverse QoS preferences of various traffic are supported within a collaborative congestion control framework. A novel multi-level, class-based scheduling and forwarding mechanism is designed to ensure varied and fine-grained QoS guarantees. A distributed congestion notification and precise feedback mechanism is also provided, which efficiently collaborates with an adaptive multipath forwarding strategy and consumer rate adjustment to rationally allocate network resources and improve transmission efficiency, particularly in multipath scenarios. Extensive experimental results demonstrate that QSCCP satisfies diverse QoS requirements while achieving outstanding transmission performance. It outperforms existing schemes in throughput, fairness, delay, and packet loss, with a rapid convergence rate and excellent stability.**

***Index Terms*—Information-Centric Networks, Congestion Control, Transport Protocol, Quality of Service (QoS).**

I. INTRODUCTION

Internet usage has been dominated by content retrieval and distribution. Due to the exponential rise of content-based services, the traditional host-based network communication architecture cannot meet the enormous content delivery requirements [1]. To provide an efficient content delivery solution, Information-Centric Networking (ICN) has been proposed to shift the host-centric network paradigm to a content-oriented one [2]. This novel communication mode unbinds the content and host address, which significantly simplifies content distribution and conforms to the current Internet evolution trend.

Over the past decade, numerous ICN-related projects have been extensively researched, such as Named Data Networking (NDN) [3] originating from Content-Centric Networking (CCN) [1], Publish Subscribe Internet Technology (PURSUIT) [4], and Network of Information (NetInf) [5]. We focus our attention on the NDN architecture, which is recognized as the most popular ICN paradigm. In NDN [3], a user requests content by issuing an Interest packet(s) with the content name. Each NDN router performs name-based forwarding. Any node (in-network cache or content producer) that holds the requested data can respond to the Interest packet (referred to as *Interest*) by returning the corresponding Data packet (referred to as *Data*). With these outstanding designs, NDN brings numerous benefits to network communication (e.g., flexible multipath forwarding, in-network caching, and robust mobility support) [6]. It has been adopted to enhance communication quality in various scenarios like the Internet of Things (IoT), Internet of Vehicles (IoV), and Smart Cities.

The novel transmission characteristics of NDN present new challenges for congestion control. In NDN, content can be retrieved from multiple sources, including original content repositories and caches of intermediate routers. Since different content sources may lead to varying retrieval delays, traditional Round-Trip Time (RTT)-based timeouts become unreliable indicators of network congestion in NDN [7]. Rather than inferring congestion at the consumer by monitoring RTT or packet loss, a common approach [7]–[9] is to detect congestion at intermediate nodes and notify this congestion information to the consumer via returned Data packets. This method provides explicit congestion feedback as a reliable congestion indicator, facilitating more effective handling of potential congestion,

---

This work was supported by Peking University - China Mobile Internet Joint Laboratory Project (No.R2442101010147U), China Environment for Network Innovation (CENI) (GJFGW[2020]No.386, SZFGW[2019]No.261), Guangdong Basic and Applied Basic Research Foundation (No.2022A1515010836), and Shenzhen Science and Technology Program (No.JCYJ20220531093206015, No.JCYJ20210324122013036). *(Corresponding author: Hui Li.)*

He Bai, Hui Li, Jianming Que, and Abla Smahi are with the Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, School of Electronic and Computer Engineering, Peking University, Shenzhen, China (e-mail: baihe@pku.edu.cn; lih64@pkusz.edu.cn; qjm253@pku.edu.cn; smahi_abla@pku.edu.cn). Hui Li is also with the School of Computer Science, Fuyao University of Science and Technology, Fuzhou, China (e-mail: 202401310009@fyust.org.cn).

Minglong Zhang is with the Department of Electrical and Computer Engineering, Mississippi State University, USA (e-mail: mz354@msstate.edu).

Peter Han Joo Chong is with the Department of Electrical and Electronic Engineering, Auckland University of Technology, Auckland, New Zealand (e-mail: peter.chong@aut.ac.nz).

Shuo-Yen Robert Li is with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu, China (e-mail: bobli@uestc.edu.cn).

Xiyu Wang and Ping Lu are with ZTE Corporation, Shenzhen, China (e-mail: wang.xiyu@zte.com.cn; lu.ping@zte.com.cn).

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

thereby improving transmission efficiency. Nevertheless, a key challenge remains in how to achieve more appropriate congestion feedback and efficient notification while improving transmission performance and resource allocation in complex multipath transmission scenarios in NDN.

Improving transmission performance in multipath scenarios is a critical and ongoing focus of NDN congestion control research. Built upon a powerful *stateful forwarding plane/engine* [10], NDN's built-in support for multipath content delivery allows it to fully utilize network resources and reduce content retrieval time [11]. To address the issue of mixed RTT measurements from different content sources, earlier studies [12], [13] introduced the concept of route labels, which are tagged in each Data packet to specify its source node and the path it traversed. Recent works (e.g., [14]–[17]) have also considered path label-based mechanisms in their designs. These works make consumers responsible for managing multiple transmission paths. Based on this, a path tag/label is added to each Interest packet, using these path labels to guide the forwarding of Interest packets. However, concerns have been raised about the feasibility and scalability of path labels [18], since the number of potential routes grows exponentially with the increase in network nodes [7]. This approach may also restrict the flexibility of NDN's native name-based adaptive multipath content retrieval [19], especially in complex and dynamically changing network conditions. In summary, despite existing congestion control studies for NDN having made many efforts, transmission performance, especially in multipath scenarios, still requires further improvement to address the aforementioned concerns.

On the other hand, most existing congestion control schemes in ICN do not consider the diverse requirements of different applications. Actually, various emerging applications in modern networks have very distinct Quality of Service (QoS) requirements in key performance metrics such as throughput, latency, packet loss, and jitter [20]. For example, delay-sensitive real-time interactive applications, such as autonomous driving and cloud gaming, often require low latency in the range of a few milliseconds. And throughput-sensitive applications, like file sharing, typically prioritize high bandwidth. Therefore, the protocol design of the underlying congestion control in ICN should not only focus on enhancing network transmission performance but also take a step further to perceive and meet the diverse QoS requirements of various applications.

In light of the analyses above, in this paper, we seek to design an NDN congestion control solution that simultaneously satisfies two key objectives: (1) to support and accommodate the diverse QoS preferences of a wide range of modern network applications during the congestion control process; (2) to further enhance overall transmission performance, improving content retrieval efficiency and ensuring reasonable resource allocation, especially in multipath scenarios.

To achieve the aforementioned objectives, we propose QSCCP, a novel QoS-aware Congestion Control Protocol based on NDN. QSCCP provides broader QoS guarantees for various applications while enhancing transmission efficiency with improved effectiveness and flexibility for content retrieval and resource allocation, especially in multipath scenarios.

The key contributions of this paper are summarized as follows:

- We present a novel QoS-aware congestion control protocol (QSCCP) implemented on NDN, a mainstream ICN paradigm. QSCCP establishes a collaborative congestion control framework that accommodates a broad range of QoS preferences while enhancing overall transmission efficiency through the efficient coordination of intermediate nodes and consumers. To the best of our knowledge, QSCCP is the first solution in the NDN domain to comprehensively address efficient congestion control while providing diverse and flexible QoS guarantees.
- QSCCP integrates diverse QoS support into the NDN congestion control process. It introduces a novel multi-level, tree-structured scheduling mechanism to manage passing traffic at intermediate nodes. In this way, this mechanism effectively mitigates or prevents network congestion while meeting the fine-grained QoS preferences (e.g., bandwidth, latency, packet loss) of various NDN traffic, such as delay-sensitive and delay-tolerant applications, and offers good scalability.
- QSCCP offers a distributed congestion notification mechanism to achieve efficient congestion feedback, especially in multipath scenarios. It leverages in-flight NDN packets to piggyback accurate feedback. Based on this feedback information, intermediate nodes along the transmission path can optimize resource allocation and convey the latest congestion information about available resources to downstream nodes through returned Data packets. This information is utilized to guide adaptive multipath forwarding at NDN router and efficient sending rate adjustment at the consumer accordingly.
- We conduct extensive performance evaluations to validate the effectiveness and feasibility of QSCCP on both a simulation platform based on NS-3 (see Section V) and an emulation platform based on Mininet (see Appendix A). Extensive experimental results demonstrate that, in comparison to existing solutions, QSCCP not only provides diverse and reliable QoS guarantees but also achieves consistent high performance, in terms of convergence rate, fairness, and stability. We have open-sourced the code [21] of our proposed scheme to facilitate further exploration within the ICN community.

The rest of this paper is organized as follows. Section II introduces the related works and existing problems. Section III outlines the design principles of our solution. Section IV describes the design details of QSCCP. Section V presents extensive performance evaluations and analyzes the results. Section VI concludes this paper with an outlook.

II. RELATED WORK

Existing congestion control solutions in ICN networks, particularly in NDN, can be categorized into three groups: 1) receiver-driven methods, 2) hop-by-hop methods, and 3) hybrid methods.

In receiver-driven methods, the receiver adjusts its Interest sending rate based on congestion signals, such as packet loss,

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

RTT variation, and explicit congestion notification. To address the issue of unreliable timeout estimation in the context of the intrinsic multi-source and multipath content delivery, CCTCP (Content Centric TCP) [22] adopts *anticipated Interests* to predict the locations of data chunks before they are requested, and maintains an individual RTO for each expected source at the receiver. OMCC-RF (Optimal Multipath Congestion Control with Request Forwarding) [13] is another representative solution. It realizes separate RTT monitoring for each route via the route label carried in each Data packet as proposed in [12], and adjusts its sending rate through an AIMD-based window adjustment mechanism. Besides, it combines a dynamic request forwarding algorithm to select the optimal Interest forwarding interface at intermediate nodes. Recent studies [15]–[17], [23] allow the receiver to decide the forwarding paths for Interest packets. Given this, they execute independent congestion control and path-specified forwarding along each transmission path of a multipath flow. However, this method restricts flexible content retrieval in NDN. Additionally, the pure receiver-driven methods are incapable of coping with the misbehaving client's incessantly excessive requesting rate. As described in Section IV, in contrast, our proposed scheme employs a collaborative and adaptive approach to resolve these challenges.

Hop-by-hop methods restrict the forwarding of Interest packets at intermediate nodes. Due to the one-Interest-one-Data transmission mode of NDN, network congestion caused by Data packets can be prevented by dropping or delaying Interest packets early at intermediate nodes [24]–[26]. Also, the intermediate nodes can divert the traffic that exceeds the link capacity to alternative paths. HIS (Hop-by-hop Interest Shaping) [27] formulates the Interest shaping as an optimization problem, and pre-calculates an optimal Interest forwarding rate using local link information. MIRCC (Multipath-aware ICN Rate-based Congestion Control) [14], inspired by the classic IP RCP algorithm, is a representative rate-based approach. At each intermediate node, it employs a dual-class rate management scheme: a primary rate to achieve max-min fairness among flows, and a secondary rate to maximize network utilization. Another well-known rate-based approach is IRNA (Interest Rate Notification and Adjustment) [28], [29]. It enables routers to specify the appropriate Interest rates of downstream nodes, and achieves per-flow fairness using a cooperative rate adjustment mechanism. In brief, intermediate nodes in NDN are uniquely positioned to restrict passing traffic or notify other nodes of congestion information. This motivates us to harness their benefits to improve transmission efficiency.

Hybrid methods combine receiver-driven and hop-by-hop congestion control. As a state-of-the-art hybrid scheme, PCON (Practical Congestion Control) [7] detects link congestion by measuring packet queuing time, and signals it towards consumers in an ECN-like manner. Downstream routers divert Interest traffic to uncongested paths by a multipath forwarding strategy, and the consumer adjusts its Interest sending rate using a BIC-based algorithm. Through our evaluations, which are detailed in Section V, we observed that this scheme has a relatively slow convergence rate and poor stability. Besides, some works [16], [30] handle congestion control as a global optimization problem. ACCP (Adaptive Congestion Control Protocol) [31] utilizes deep learning techniques to predict network congestion at the intermediate node and returns it back to the receiver. However, this approach imposes considerable computation cost on in-network nodes. Also, its generalization in varying network conditions remains to be investigated.

On the other hand, previous congestion control mechanisms in NDN have failed to incorporate QoS guarantees into their protocol designs, which are essential for meeting the diverse performance requirements of modern network applications. There is a wealth of experience with QoS design in IP networks, such as the well-known IntServ model [32] and DiffServ model [33]. Although these models have not been widely deployed, the efforts they made have provided lessons for the development of the QoS-related research field. In fact, compared to host-centric IP networks, NDN has greater potential for providing more flexible, diverse, and scalable QoS support [34]. This potential is attributed to the novel characteristics of the NDN architecture [3], such as the stateful forwarding plane, symmetric routing, flows grouped by name prefix, and inherent flow balance. As a promising future network architecture, NDN is still in an evolving phase. Its native design, such as its protocols [1], [3], packet format [35], and forwarder [10], emphasizes flexibility and scalability. This provides researchers with the opportunity to further explore new functionalities and applications, fully realizing NDN's potential and driving its continued development.

Recent studies have explored the integration of QoS into the architecture and protocols of ICN networks. For instance, Kim *et al.* [36] have proposed a simplified ICN DiffServ model based on the IP DiffServ architecture [33]. Wang *et al.* [37] have developed an energy-efficient ICN QoS routing mechanism that considers QoS factors from bandwidth, delay and error rate to enhance content retrieval efficiency. Focusing on the QoS requirements for a specific ICN communication scenario, particularly within constrained ICN Internet of Things (IoT) networks, Gündoğan *et al.* [38] have presented a distributed resource management strategy with four service classes based on *latency* and *reliability*. These efforts indicate that, driven by emerging applications with varying preferences for network performance, the continuously evolving ICN protocols are advancing toward the incorporation of QoS functionalities. Therefore, in the context of ICN congestion control research, it is crucial to account for QoS-related support.

As mentioned earlier, most existing NDN congestion control studies do not account for QoS-related support. In [39], we explored a priority-based congestion control scheme named DSCCP for ICN. The core goal of this preliminary work was to achieve coarse-grained bandwidth allocation based on service priorities, without considering or incorporating a comprehensive, fine-grained QoS design. To bridge the research gap of providing diverse QoS support within NDN congestion control, one of QSCCP's key design objectives is to meet a wide range of diverse QoS requirements of various modern network applications in the NDN congestion control process. Based on this, QSCCP not only focuses on mitigating and preventing network congestion but also emphasizes

QoS-aware fine-grained resource allocation. It is capable of handling the diverse requirements of various latency-sensitive and latency-tolerant traffic, taking into account their specific QoS preferences regarding bandwidth, latency, and other performance metrics. In addition to the diverse QoS-related concerns mentioned above, QSCCP also focuses on improving the transmission efficiency of NDN congestion control itself. Therefore, another key design objective of QSCCP is to improve the transmission performance of NDN with reasonable resource allocation, especially in multipath scenarios. DSCCP does not consider this aspect, making it difficult to efficiently utilize link resources in multipath scenarios, which may potentially result in significant performance degradation. This design objective is also an important direction that existing NDN congestion control studies need to further enhance, as discussed earlier in this section.

In summary, based on the above discussions, while existing congestion control solutions in ICN networks have made significant strides in improving content retrieval efficiency, there remain numerous unresolved issues. Our work seeks to enhance transmission performance and address the limitations of existing solutions, while also taking a further step to incorporate broader QoS concerns.

III. DESIGN PRINCIPLES

The proposed protocol in this paper aims at enabling new perspectives for efficient congestion control in ICN networks, while also integrating the diverse QoS guarantee to support different performance requirements of various ICN applications. In light of the fact that ICN is still in its early stages of development, we place a greater emphasis on maximizing the ICN's potential through openness and design flexibility. We choose the NDN architecture to implement our scheme. The design principles are as follows.

1) Meet the diverse QoS preferences: Numerous modern network applications have diverse QoS requirements (e.g., throughput, delay, and packet loss). A live baseball game viewed by millions of spectators, for instance, has stringent requirements on low latency and high packet delivery rate. Existing ICN transport protocols lack effective and comprehensive QoS support, as discussed in Section II. To bridge this gap, QSCCP empowers the congestion control framework built on NDN to effectively handle the diverse QoS preferences of various applications. QoS guarantees are provided throughout the entire network transmission. A set of suggested NDN service classes is defined, with support for compatibility and scalability. QSCCP incorporates QoS-related support into traffic scheduling and packet forwarding. It implements a novel multi-level, class-based scheduling mechanism to meet the fine-grained and diverse QoS preferences of various NDN traffic. Based on the collaboration between intermediate nodes and the consumer through each module, QSCCP enables QoS-aware rational resource allocation while effectively mitigating or preventing network congestion.

2) Leverage intermediate nodes to assist in congestion control with QoS support: Traditional congestion control primarily operates on the consumer side, whereas routers are responsible for fundamental forwarding and routing. Encouragingly, the NDN architecture introduces a powerful stateful forwarding engine [3] that employs novel data structures (i.e., the PIT (Pending Interest Table), FIB (Forwarding Information Base), and CS (Content Store)). This forwarding engine enhances the processing capability of the router and provides adaptable forwarding [10]. Drawing inspiration from this concept, QSCCP takes advantage of intermediate nodes to assist in managing various traffic with varied performance preferences. Through class-based traffic scheduling, adaptive multipath forwarding, and precise signaling and feedback mechanisms at intermediate nodes, in collaboration with rate adjustment on the consumer side, our scheme can achieve efficient congestion control while providing diverse QoS-related support.

3) Remove inefficient AIMD-style window adjustment approach: The traditional AIMD-style window adjustment approach is not ideal for ICN networks [14]. For instance, this approach consumes multiple RTTs to implicitly probe the proper sending window, which leads to a prolonged response time and poor stability. Moreover, it tends to be overly sensitive in handling burst traffic, as shown by experimental findings in Section V. These observed phenomena also stem from the fact that the RTT-sensitive AIMD-style window adjustment mechanisms fail to effectively identify the bottleneck structure of the network [40], [41], especially in multipath transmission scenarios. Instead, QSCCP implements a rate controller at the consumer and adjust the Interest sending rate based on the accurate feedback inside the received Data packets. Through this approach, the consumer can swiftly find a suitable operating point within a few RTTs, effectively reducing flow completion time.

4) Enable efficient congestion notification and resource allocation for multipath flows: The multipath transmission property has significant advantages for content delivery in NDN. Nevertheless, how to efficiently execute congestion control and resource allocation in multipath scenarios remains a persistent and critical challenge. A common solution [14]–[17] is to manage transmission paths at the consumer side and perform path label-based packet forwarding. However, this approach faces limitations in terms of practicality, flexibility, and scalability, as discussed in Section I. Unlike this approach, QSCCP leverages precise feedback to provide more useful guidance for congestion control in multipath scenarios. In-flight NDN packets carry and convey feedback information to intermediate nodes they pass through. By utilizing the received feedback information and local congestion status, intermediate nodes can optimize resource allocation and encapsulate accurate congestion information regarding the latest available resources into returned Data packets. This information is then used to guide adjustments for both the consumer's Interest sending and the forwarding strategy. Under this distributed and collaborative mechanism, QSCCP enhances transmission performance in multipath scenarios and achieves rational resource allocation. The design details are discussed in Section IV-D.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

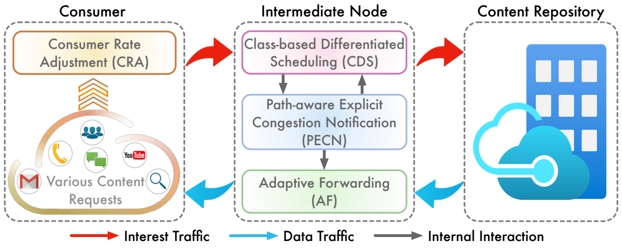


Fig. 1. Overview of the QSCCP architecture.

IV. QoS-AWARE CONGESTION CONTROL PROTOCOL

Building upon the design principles outlined in Section III, we propose a QoS-aware congestion control protocol (QSCCP) based on NDN to provide diverse and flexible QoS guarantees while enhancing overall transmission efficiency and resource allocation, especially in multipath scenarios. The following subsections detail the proposed protocol.

A. QSCCP Overview

The overall architecture of the proposed QSCCP protocol is depicted in Fig. 1. The QSCCP consists of four modules: (1) Class-based Differentiated Scheduling (CDS), (2) Path-aware Explicit Congestion Notification (PECN), (3) Adaptive Forwarding (AF), and (4) Consumer Rate Adjustment (CRA). These modules function in tandem with one another. The first three logical modules are added at intermediate nodes, as shown in Fig. 2. For varying incoming Interest traffic, the CDS module performs a multi-level, class-based scheduling based on the service class indicated in each received Interest packet. The PECN module estimates the available resources of passing flows and optimizes resource allocation, especially for multipath flows. It conveys instant, precise feedback information via the returned Data packets to downstream nodes. Based on this feedback information, the AF module adjusts the Interest forwarding probability of each available interface to provide load balancing. The CRA module, located at the consumer, adjusts the Interest sending rate according to the feedback information in the received Data packets. The key notations used in our design are summarized in Table I for clarity.

B. Packet Format and Service Class

As illustrated in Fig. 3, QSCCP introduces several new fields into NDN packets. Notably, for the evolving NDN architecture, the reasonable addition of new fields to NDN packets is permitted, allowing researchers to explore new protocol designs and functionalities to advance the development of NDN. The NDN packet is encoded in a Type-Length-Value (TLV) format, which has excellent compatibility and scalability [35]. Note that in our scheme, the additional fields are excluded from the packet signature calculation. Our operations regarding the introduction of these new fields do not disrupt existing operations or violate the core principles and functions of the NDN architecture. A brief description of their functions is provided below:

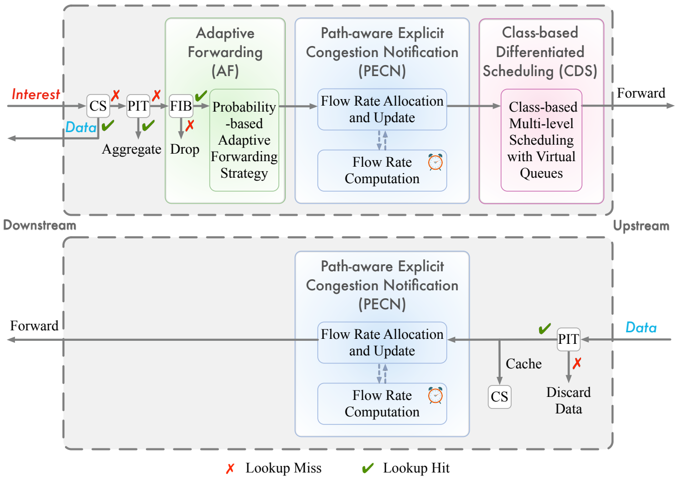


Fig. 2. Interest and Data processing at intermediate nodes in QSCCP.

TABLE I
LIST OF NOTATIONS AND DESCRIPTIONS

Notation	Description
$\mathbb{C}$	The set of service classes
$\mathbb{C}_{DS}$	The subset of service classes for delay-sensitive traffic
$\mathbb{C}_{DT}$	The subset of service classes for delay-tolerant traffic
$\mathbb{P}$	The set of prefixes of flows passing through a particular router interface
$\mathbb{P}_c$	The subset of prefixes for passing flows with the service class c via a certain router interface
$R(t)$	The suggested target rate of a flow
η	The target link utilization
$B(t)$	The link capacity
$B'(t)$	The available link capacity reserved for returned Data packets
δ_c	The limited total available link capacity (percentage) for the delay-sensitive traffic with service class c at a certain interface
w_c^p	The bottom level scheduling weight of a flow with the service class c and the prefix p
ω_c	The middle level scheduling weight of the delay-tolerant traffic with the service class c
$\varphi_c^p(t)$	The discount factor of a flow with the service class c and the prefix p
$(f_c^p)_j$	The forwarding probability of a flow with the service class c and the prefix p at the interface j
$(d_c^p)_i$	The estimated downstream bottleneck rate of a flow with the service class c and the prefix p at the interface i
$(u_c^p)_k$	The estimated integrated upstream bottleneck rate of a flow with the service class c and the prefix p at the interface k

- *ServiceClass*: This field indicates the service class of a content request. In QSCCP, each NDN flow is identified by the name prefix and associated service class. Inspired by the common classification used in IP networks [33], the suggested NDN service classes are as follows:
  - CS (Class Selector): includes CS7 and CS6, dedicated to the network control messages, such as routing protocols.
  - EF (Expedited Forwarding): provides low delay, low jitter, low loss, and a guaranteed amount of bandwidth for time-sensitive traffic, such as voice, video conference, and other real-time services.
  - AF (Assured Forwarding): includes four separate AF classes for queuing purposes, i.e., AF4, AF3,

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

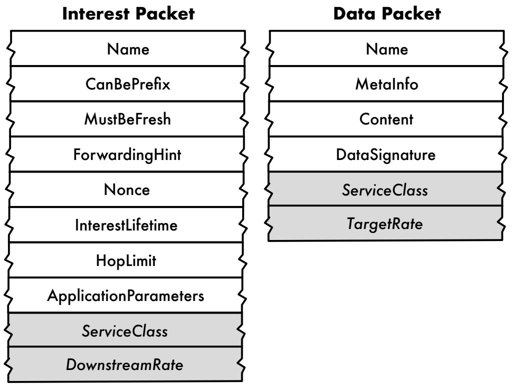


Fig. 3. The packet format in QSCCP (The gray shaded areas represent the new fields).

  AF2, and AF1, each affording different forwarding assurances for content delivery under prescribed conditions.
  - BE (Best Effort): the default service class.

  The suggested NDN service classes offer a foundational classification solution that allows for further customization or expansion based on specific requirements. Based on the defined service classes, QSCCP integrates QoS-related support into the NDN congestion control process, as detailed in Sections IV-C and IV-D. It is worth noting that alternative reasonable definitions or implementations (e.g., incorporating the NDN service class as part of the name component) are also permissible and compatible with the proposed QSCCP framework.
- *DownstreamRate*: This field inside the Interest packet conveys the estimated downstream bottleneck rate (in bits per second) of a flow between the passing node and the consumer. Upon receiving an Interest packet, the intermediate node records and updates the value of this field to facilitate the subsequent updating of *TargetRate* field inside the Data packet. Further details are elaborated in Section IV-D.
- *TargetRate*: This field inside the Data packet is recorded and updated by the intermediate node to transmit the estimated *suggested target rate* (in bits per second) of the flow to the consumer. This process is described in detail in Section IV-D. Additionally, the recorded values are utilized in the local calculation of Interest forwarding probability, as demonstrated in Section IV-E.

C. Class-based Differentiated Scheduling

As stated previously, the NDN data transmission mode is one-Interest-one-Data. By regulating the Interest traffic at intermediate nodes, the returned Data traffic can be effectively constrained in advance [11]. Given this, we design the Class-based Differentiated Scheduling (CDS) module to accomplish class-based scheduling and forwarding for diverse traffic with different QoS preferences and to prevent network congestion.

As depicted in Fig. 4, the CDS module consists of a Multi-level Scheduler (MS) and multiple Token-Bucket Rate Limiters (TBRLs). At each outgoing interface of an intermediate node, the CDS module maintains one separate virtual queue for each passing flow, identified by the name prefix and the marked service class. The MS manages and schedules virtual queues using a three-level, tree-structured scheduling paradigm. When an Interest packet arrives at an outgoing interface, the CDS module checks the packet's service class and name prefix in order to allocate it to the appropriate virtual queue. Then, the CDS module utilizes the MS to execute class-based packet scheduling, while multiple TBRLs are utilized to limit the forwarding of Interest packets. The following are the design specifications of the CDS module.

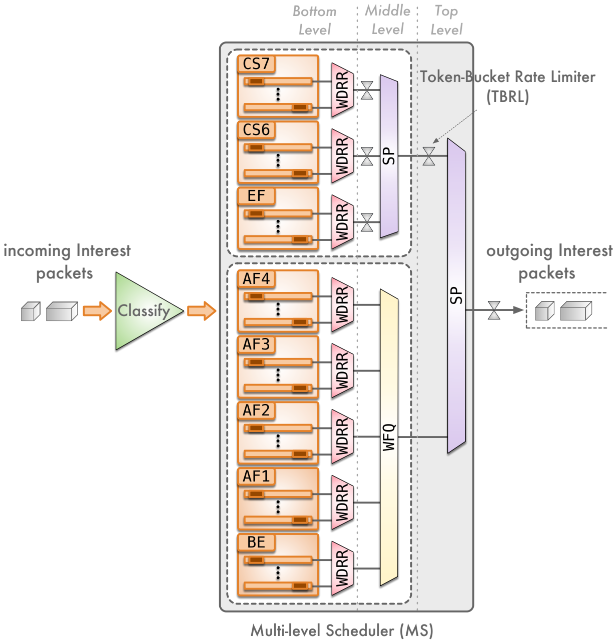


Fig. 4. Class-based Differentiated Scheduling module.

1) Top Level Scheduling: The CDS module performs strict priority (SP) scheduling between delay-sensitive traffic and delay-tolerant traffic. Delay-sensitive traffic (i.e., CS7, CS6, and EF) usually has stringent delay requirement; hence, it should be served prior to delay-tolerant traffic (i.e., AF, BE). TBRLs are used to limit the bandwidth consumption of the delay-sensitive traffic. As a result, the potential greedy behavior of delay-sensitive traffic, such as overprioritizing or draining link bandwidth, which may starve the delay-tolerant traffic, can be effectively avoided. The specific parameters of the TBRLs can be configurable flexibly by network operators.

2) Middle Level Scheduling: The CDS module performs SP scheduling for delay-sensitive traffic and weighted fair queuing (WFQ) scheduling for delay-tolerant traffic. Generally, different types of delay-sensitive traffic have distinct delay requirements. Therefore, it is necessary to execute the SP scheduling in accordance with their respective priority orders, namely, CS7, CS6, and EF. Also, their individual packet forwarding is limited by the TBRLs. In contrast to delay-sensitive traffic, delay-tolerant traffic (i.e., AF, BE) does not have strict delay requirements. Thus, a WFQ-like scheduling algorithm is executed among them to guarantee the requisite bandwidth and controllable delay, as well as to balance the

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

overall jitter among different types of delay-tolerant traffic. This scheduling mode achieves the weighted max-min fair bandwidth allocation among delay-tolerant traffic with different service classes. Moreover, if a queue is empty, other queues can share the idle bandwidth proportionally to their respective weights, maximizing link utilization. It ensures that all types of traffic can obtain the minimum bandwidth quota and that no queue is starved, even if the link is congested.

3) Bottom Level Scheduling: The CDS module performs weighted deficit round robin (WDRR) scheduling among flows with the same service class. This enables weighted bandwidth allocation among flows with the same service class and maintains a minimum scheduling overhead of $O(1)$, which does not scale with the number of flows. Furthermore, TBRL can also be applied to a single delay-sensitive flow if the potential traffic policies configured by network operators specify related requirements. During this scheduling process, it should be noted that the *greedy flow* should be penalized. This is due to the potential for the network bandwidth to be significantly occupied by this greedy flow when a misbehaving consumer maintains a high constant Interest sending rate and refuses to make the necessary adjustments required by a consumer-side congestion control mechanism. Fortunately, the NDN router can limit the greedy flow [11]. In QSCCP, if a flow is judged in a greedy status, its bottom level scheduling weight (i.e., its WDRR scheduling weight) will be discounted by a discount factor. As a result, the actual bandwidth allocated to this greedy flow will be less than intended. This enforces the misbehaving consumer to resume normal sending state. We use φ to represent the discount factor that indicates the punishment intensity, $\varphi \in (0, 1]$. The default initial value of φ is 1, which indicates no punishment. More details are introduced in Section IV-D.

Using the above design, the CDS module can satisfy different QoS requirements (e.g., delay, bandwidth) of various traffic with low scheduling overhead and computation complexity of $O(1)$. This shows that our scheme is highly feasible. In addition, following the MS scheduling, a TBRL is applied to limit the total amount of Interest traffic forwarded through an outgoing interface. The committed information rate (CIR) of this TBRL is set as the product of the link capacity and target link utilization. This ensures that the returned Data traffic will not exceed the available link capacity.

D. Path-aware Explicit Congestion Notification

The optimal sending rate at the consumer is not a priori value, but changes depending on the variable network conditions [11]. Thus, the consumer needs to obtain real-time network status in order to adjust its Interest sending rate in a reasonable manner. In this subsection, we design the Path-aware Explicit Congestion Notification (PECN) module, which provides precise feedback information about available network resources and optimizes resource allocation to efficiently handle multipath scenarios. To maximize feedback efficiency, we utilize the estimated *suggested target rate* of a flow, denoted as $R(t)$, as the feedback information piggybacked on the returned Data packet through the *TargetRate* field. The flow rate computation, allocation, and updating are described below.

1) Flow Rate Computation: The PECN module periodically estimates a *suggested target rate* for the passing flow. In the CDS module, delay-sensitive traffic and delay-tolerant traffic are scheduled using a hierarchical approach, thereby their rates should be estimated separately. The computations are detailed below.

Specifically, we assume that $\mathbb{C} = \{c \mid c \in \mathbb{C}_{DS}$, or $c \in \mathbb{C}_{DT}\}$ denotes the set of service classes as described in Section IV-B. $\mathbb{C}_{DS} = \{\mathrm{CS7}, \mathrm{CS6}, \mathrm{EF}\}$ denotes the subset of service classes for delay-sensitive traffic, and $\mathbb{C}_{DT} = \{\mathrm{AF4}, \mathrm{AF3}, \mathrm{AF2}, \mathrm{AF1}, \mathrm{BE}\}$ denotes the subset of service classes for delay-tolerant traffic. We further assume that $\mathbb{P}$ is the set of prefixes of flows passing through a particular router interface. $\mathbb{P}_c$ represents the subset of prefixes of passing flows with the service class c via a certain router interface. Besides, to enhance the accuracy of flow rate computation, the link capacity occupied by Interest packets in the reverse direction is no longer negligible [27]. In practice, the available capacity reserved for returned Data packets at an outgoing interface of a router is calculated by:

$$B'(t) = \eta \cdot B(t) - B_I(t), \tag{1}$$

where $B(t)$ is the link capacity, η is the target link utilization which allows the network to target a peak link utilization of less than 100% under overload conditions, and $B_I(t)$ is the measured arriving rate of all Interest traffic in the reverse direction at this interface.

On the basis of the preceding definitions, we can derive the main elements in the flow rate computation. Note that all rate elements mentioned below refer to the expected rate of returned Data packets, unless otherwise specified.

First, we define the weighted fair rate of the passing flow, allocated by the hierarchical scheduling of the CDS module. For a certain outgoing interface of a router, the weighted fair rate of a delay-sensitive flow with the service class c ($c \in \mathbb{C}_{DS}$) and the prefix p ($p \in \mathbb{P}_c$), is defined as follows:

$$base\_rate_c^p(t) = B'(t) \cdot \delta_c \cdot \frac{w_c^p \cdot \varphi_c^p(t)}{\sum_{p' \in \mathbb{P}_c} w_c^{p'} \cdot \varphi_c^{p'}(t)}. \tag{2}$$

Here, δ_c represents the limited total available link capacity (percentage) for the delay-sensitive traffic with service class c at this interface. w_c^p is the bottom level scheduling weight of this delay-sensitive flow, allocated by the WDRR scheduling in the CDS module. $\varphi_c^p(t)$ is the discount factor of a flow, as introduced in Section IV-C. For a penalized greedy flow, $\varphi_c^p(t) \in (0, 1)$, while for other flows (i.e., the later-introduced high-demand flow and the low-demand flow), $\varphi_c^p(t)$ is equal to the default value 1. $\sum_{p' \in \mathbb{P}_c} w_c^{p'} \cdot \varphi_c^{p'}(t)$ denotes the sum of the products of the bottom level scheduling weight and the corresponding discount factor of each delay-sensitive flow with the service class c via this interface. The following formula defines the weighted fair rate of a delay-tolerant flow with the service class c ($c \in \mathbb{C}_{DT}$) and the prefix p ($p \in \mathbb{P}_c$):

$$\begin{aligned} base\_rate_c^p(t) = &(B'(t) - total\_rate_{DS}(t)) \\ &\cdot \frac{\omega_c}{\sum_{c' \in \mathbb{C}_{DT}} \omega_{c'}} \cdot \frac{w_c^p \cdot \varphi_c^p(t)}{\sum_{p' \in \mathbb{P}_c} w_c^{p'} \cdot \varphi_c^{p'}(t)}. \end{aligned} \tag{3}$$

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

The $total\_rate_{DS}(t)$ here refers to the sum of the expected Data returning rates inflated by measured actual Interest forwarding rates of delay-sensitive flows via this interface. ω_c is the middle level scheduling weight of the delay-tolerant traffic with the service class c, allocated by the WFQ scheduling in the CDS module, and $\sum_{c' \in \mathbb{C}_{DT}} \omega_{c'}$ is the sum of the middle level scheduling weights of the passing delay-tolerant traffic. w_c^p is the bottom level scheduling weight of this delay-tolerant flow, and $\sum_{p' \in \mathbb{P}_c} w_c^{p'} \cdot \varphi_c^{p'}(t)$ denotes the sum of the products of the bottom level scheduling weight and the corresponding discount factor for each delay-tolerant flow with the service class c via this interface.

Then, we define the *flow states* to assist intermediate nodes in estimating the suitable suggested target rate for the passing flow individually based on its actual network resource demand. We specify the $arrive\_rate_c^p(t)$ to denote the inflated Interest arriving rate of a flow, which is periodically measured by the PECN module and inflated to the expected Data returning rate. There are three distinct types of *flow states*: *greedy state* (G), *high-demand state* (H), and *low-demand state* (L). Specifically, the *greedy state* indicates the greedy behavior of a misbehaving consumer who maintains a high constant Interest sending rate and does not cooperate with the consumer rate adjustment, as mentioned in Section IV-C. A flow is judged to be in a *greedy state*, if, for λ consecutive times, its inflated Interest arriving rate exceeds its theoretically suggested target rate recorded at the local router. In this case, the CDS module will punish the greedy flow by setting its discount factor $\varphi_c^p(t)$ to a value less than 1. Note that this punishment tactic goes into effect during the subsequent flow rate computation period. For a non-greedy flow with the service class c ($c \in \mathbb{C}$) and the prefix p ($p \in \mathbb{P}_c$), the flow is in a *high-demand state* if $arrive\_rate_c^p(t) > \gamma \cdot base\_rate_c^p(t)$, where γ is a control parameter designed to absorb the rate fluctuation, otherwise, it is in a *low-demand state*. The low-demand flow typically has small resource occupation because of its application characteristics. To improve link utilization, the CDS module will reallocate idle network resources from low-demand flows to high-demand flows with greater resource demands.

According to the above flow state definitions, the *suggested target rate* of a flow via a specific outgoing interface of the router can be determined as follows. For a greedy flow with the service class c ($c \in \mathbb{C}$), the prefix p ($p \in \mathbb{P}_c$), and the discount factor $\varphi_c^p(t)$ ($\varphi_c^p(t) \in (0, 1)$), the *suggested target rate* is equivalent to its discounted weighted fair rate, namely:

$$R_G(t) = base\_rate_c^p(t). \tag{4}$$

For a low-demand flow with the service class c ($c \in \mathbb{C}$), the prefix p ($p \in \mathbb{P}_c$), and the discount factor $\varphi_c^p(t)$ ($\varphi_c^p(t) = 1$), its *suggested target rate* equals its weighted fair rate, namely:

$$R_L(t) = base\_rate_c^p(t). \tag{5}$$

As previously stated, the suggested target rate for a high-demand flow consists of its weighted fair rate and the idle bandwidth allocated to it in proportion to its weight. For delay-sensitive traffic and delay-tolerant traffic, the suggested target rate computations for their respective high-demand flows are differentiated. First, for delay-sensitive traffic with the service class c ($c \in \mathbb{C}_{DS}$), the total remaining bandwidth is:

$$\begin{aligned} remain\_rate_c(t) = B'(t) \cdot \delta_c &- \sum_{p' \in P_{c-G}} base\_rate_c^{p'}(t) \\ &- \sum_{p' \in P_{c-H}} base\_rate_c^{p'}(t), \\ &- \sum_{p' \in P_{c-L}} arrive\_rate_c^{p'}(t) \end{aligned} \tag{6}$$

where P_{c-G} is the set of prefixes of delay-sensitive flows in greedy state with service class c via this interface, P_{c-H} is the set of prefixes of delay-sensitive flows in high-demand state with service class c, and P_{c-L} is the set of prefixes of delay-sensitive flows in low-demand state with service class c. For a delay-sensitive flow in high-demand state with service class c ($c \in \mathbb{C}_{DS}$) and the prefix p ($p \in \mathbb{P}_c$), the *suggested target rate* can be derived as follows:

$$\begin{aligned} R_{DS-H}(t) = base\_rate_c^p(t) + remain\_rate_c(t) \cdot \\ \frac{w_c^p}{\sum_{p' \in P_{c-H}} w_c^{p'}}, \end{aligned} \tag{7}$$

where $\sum_{p' \in P_{c-H}} w_c^{p'}$ represents the sum of the bottom level scheduling weights of delay-sensitive flows in high-demand state with the service class c via this interface. Note that the discount factor $\varphi_c^p(t)$ for the high-demand flow equals 1, we exclude this symbol from all equations used to estimate the high-demand flow rate. Similarly, the total remaining bandwidth for delay-tolerant traffic can be calculated by:

$$\begin{aligned} remain\_rate_{DT}(t) = &B'(t) - total\_rate_{DS}(t) \\ &- \sum_{c \in \mathbb{C}_{DT}} \sum_{p' \in P_{c-G}} base\_rate_c^{p'}(t) \\ &- \sum_{c \in \mathbb{C}_{DT}} \sum_{p' \in P_{c-H}} base\_rate_c^{p'}(t) \cdot \\ &- \sum_{c \in \mathbb{C}_{DT}} \sum_{p' \in P_{c-L}} arrive\_rate_c^{p'}(t) \end{aligned} \tag{8}$$

Thus, for a delay-tolerant flow in high-demand state with the service class c ($c \in \mathbb{C}_{DT}$) and the prefix p ($p \in \mathbb{P}_c$), its *suggested target rate* can be calculated as follows:

$$\begin{aligned} R_{DT-H}(t) = base\_rate_c^p(t) + remain\_rate_{DT}(t) \\ \cdot \frac{\omega_c}{\sum_{c' \in \mathbb{C}_{DT}} \omega_{c'}} \cdot \frac{w_c^p}{\sum_{p' \in P_{c-H}} w_c^{p'}}, \end{aligned} \tag{9}$$

where $\sum_{p' \in P_{c-H}} w_c^{p'}$ denotes the sum of the bottom level scheduling weights of delay-tolerant flows in high-demand state with the service class c via this interface.

2) Flow Rate Allocation and Update: Using the aforementioned flow rate computation equations, intermediate nodes periodically estimate the local *suggested target rates* of the passing flows. Once an Interest packet reaches the content producer, the corresponding Data packet is generated and the *TargetRate* field (which contains the suggested target rate) is initialized to MAX_RATE. When the returned Data packet arrives at a router, if the latest local suggested target rate of

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

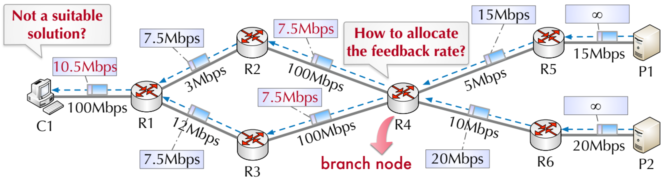


Fig. 5. Feedback rate update issue for the multipath flow (The lavender shaded area represents the value of the *TargetRate* field of a Data packet).

this flow recorded at the router is less than its current carried rate, the router will update the *TargetRate* field to the local value. Thus, once a Data packet reaches the consumer, the CRA module can obtain the available rate of this flow, and adjust the Interest sending rate proportionally to this rate. This approach enables the overall throughput to quickly attain a stable state and minimizes flow completion time.

For a *single-path flow*, at intermediate nodes, the update of carried rate for each returned Data packet conforms to the aforementioned rules. The traffic of a *multipath flow*, however, traverses multiple transmission paths, each with a distinct bottleneck. This requires the *branch nodes*, responsible for forwarding the returned Data traffic associated with the *multipath flow* to various downstream paths, to make reasonable flow rate updating decisions.

As illustrated in Fig. 5, Consumer C1 retrieves different fractions of a content object from Producer P1 and P2 via multiple transmission paths. With the current design, when the Data traffic from P1 or P2 arrives at the branch node R4, R4 knows local link states and can obtain the current bottleneck rates (from P1/P2 to R4) from the *TargetRate* field inside the received Data packets, but it cannot predict the conditions of the two downstream transmission paths (from R4 to C1) in advance. An intuitive rate update solution (Fig. 5) is that the rate value carried in each Data packet is updated to 7.5Mbps, i.e., R4 allocates the aggregate feedback rate value of this flow (i.e., 15Mbps) equally between the Data traffic waiting to be forwarded along two distinct downstream paths. Then, when the Data traffic arrives at the aggregation node R1, the rate value within each Data packet along the path R4-R2-R1 is updated to 3Mbps, the rate value within each Data packet along the path R4-R3-R1 remains 7.5Mbps, resulting in an aggregate rate value of 10.5Mbps for this multipath flow. Obviously, this solution leads to a low link utilization (the theoretical maximum reachable throughput is 15Mbps).

To address the aforementioned problem in a straightforward and efficient way, we design a novel distributed flow rate allocation and updating mechanism accommodating multipath transmission scenarios. In our scheme, the Interest packet is required to carry the estimated downstream bottleneck rate (from the arriving node to the consumer) of its transmission path through the *DownstreamRate* field. Therefore, once an Interest packet arrives at a router, the router can obtain in advance the downstream bottleneck path condition (from the current node to the consumer) of this flow from the *DownstreamRate* field inside this Interest packet. On the other hand, once a Data packet arrives at a router, the router can obtain the upstream bottleneck path condition (from the producer to the current node) of this flow from the *TargetRate* field inside this Data packet. Based on these two pieces of information and local link states, routers can easily estimate the available resources for a multipath flow on its transmission paths, thereby perform reasonable feedback rate updates as Data packets return, especially at branch nodes in multipath scenarios. The operations are explained in detail below.

When an Interest packet is issued by the consumer, its *DownstreamRate* field is initiated to MAX_RATE. Once the Interest packet arrives at an intermediate node, the current value of its *DownstreamRate* field is first recorded in a hash table called DRT (Downstream Rate Table). Then, the router updates the value of the *DownstreamRate* field inside this Interest packet. Namely, the value of the *DownstreamRate* field of an Interest packet with the service class c and the prefix p that is waiting to be forwarded from the outgoing interface j is updated to:

$$\min((R_c^p)_j(t), (f_c^p)_j \cdot \sum_{k \in I_c^p} (d_c^p)_k). \tag{10}$$

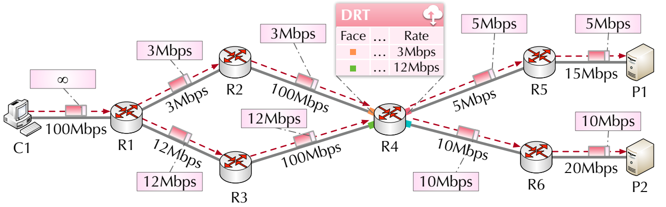
Here, $(R_c^p)_j(t)$ represents the latest local suggested target rate of this flow at the outgoing interface j, which is estimated periodically by the router, as previously stated. $(f_c^p)_j$ is the forwarding probability of this flow at the outgoing interface j, as computed by the later-introduced equation (12). I_c^p denotes the set of all recorded Interest incoming interfaces of this flow at the current node. $\sum_{k \in I_c^p} (d_c^p)_k$ is the sum of the recorded downstream bottleneck rates of this flow at its Interest incoming interfaces stored in the table DRT at the current node. The equation (10) indicates that the value of the *DownstreamRate* field inside this Interest packet is updated to the smaller one between the latest local suggested target rate of this flow at the selected Interest forwarding interface and the theoretically allocated downstream bottleneck rate of the Interest traffic that belongs to this flow and is waiting to be forwarded from the selected outgoing interface.

With downstream bottleneck information recorded in DRT at intermediate nodes, the flow rate updating for returned Data packets can be performed reasonably. This is especially true for branch nodes in multipath transmission scenarios. Upon receiving a Data packet, the router compares the value of the *TargetRate* field with the latest calculated local suggested target rate of the flow at the incoming interface of this Data packet, and stores the smaller value in a hash table called TRT (Target Rate Table). The TRT entry represents the estimated integrated upstream bottleneck rate of a flow at a certain interface, which combines the local latest and notified upstream path conditions. The router then updates the value of the *TargetRate* field inside the Data packet. Specifically, the *TargetRate* field of a Data packet with the service class c and the prefix p that is waiting to be forwarded from the outgoing interface i is updated to:

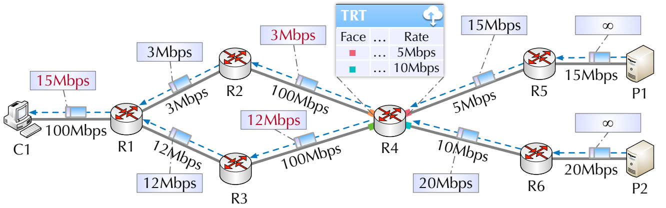
$$\frac{(d_c^p)_i}{\sum_{k \in I_c^p} (d_c^p)_k} \cdot \sum_{k \in O_c^p} (u_c^p)_k. \tag{11}$$

Here, $(d_c^p)_i$ is the estimated downstream bottleneck rate of this flow at the interface i recorded in the table DRT. I_c^p denotes

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052



(a) The per-hop update of *DownstreamRate* field during the transmission of Interest traffic (The light pink shaded area represents the value of the *DownstreamRate* field of an Interest packet).



(b) The per-hop update of *TargetRate* field during the transmission of Data traffic (The lavender shaded area represents the value of the *TargetRate* field of a Data packet).

Fig. 6. An example of feedback rate update for the multipath flow in QSCCP.

the set of Interest incoming interfaces (i.e., Data outgoing interfaces) of this flow at the current node. O_c^p denotes the set of Interest outgoing interfaces (i.e., Data incoming interfaces) of this flow at the current node. $\sum_{k\in O_c^p}(u_c^p)_k$ is the sum of the recorded integrated upstream bottleneck rates of this flow at its Data incoming interfaces contained in the table TRT at the current node. The returned Data traffic of a multipath flow may have multiple outgoing interfaces at the branch node. The calculation result of equation (11) indicates the current suggested target rate allocated to the Data traffic of this flow waiting to be forwarded from the selected outgoing interface (i.e., the original corresponding Interest incoming interface).

Using the above design, QSCCP achieves proper resource allocation and feedback rate updating for both single-path and multipath transmission scenarios. An example is depicted in Fig. 6.

One distinct feature of our design is that we do not require path labels to identify individual routes for multipath congestion control. The practicality and scalability of path label-based approaches (e.g., [12], [13], [15]–[17]) have been questioned [7], [9], [18]. Path label-based schemes require maintaining a large number of routes at the consumer side, which presents scalability and high-overhead challenges as the number of potential routes grows exponentially with the number of nodes in the network [18]. Moreover, their reliability and effectiveness under dynamic and complex network conditions remain open issues [7]. Unlike them, our solution provides precise feedback information to both the receiver and the router's forwarding strategy via in-flight NDN packets. Owing to the PECN module, this feedback carried by packets is reasonably allocated and updated at intermediate nodes, particularly at branch nodes in multipath transmission. As a result, our approach does not require actively identifying, maintaining, and managing all paths or monitoring the state of each individual path. Instead, it focuses on the information recorded at the upstream and downstream interfaces of intermediate nodes, especially at branch nodes, to handle resource allocation and feedback rate updating for multipath flows. With this design, our solution efficiently handles congestion control in multipath scenarios and adapts well to varying network conditions.

E. Adaptive Forwarding

In NDN, each router maintains a list of forwarding interfaces for each name prefix in FIB (Forwarding Information Base), and its forwarding strategy decides whether, how, and to which interface (next hop) an Interest packet is forwarded [10]. We design an Adaptive Forwarding (AF) module that contains a probability-based adaptive forwarding strategy to control traffic load and improve link utilization, especially in multipath transmission scenarios.

As part of its primary mission, the AF module selects forwarding interfaces for arriving Interest traffic based on a probability distribution. As the PECN module records local link states and path conditions, every alternative outgoing interface for the Interest traffic of a flow can be weighted by the corresponding estimated integrated upstream bottleneck rate, recorded in the previously described TRT table. Thus, the forwarding probability of an Interest packet with the service class c and prefix p at an outgoing interface j is defined by:

$$(f_c^p)_j = \frac{(u_c^p)_j}{\sum_{k\in O_c^p}(u_c^p)_k}, \tag{12}$$

where $(u_c^p)_j$ is the estimated integrated upstream bottleneck rate of this flow at the interface j stored in the table TRT, and $\sum_{k\in O_c^p}(u_c^p)_k$ is the sum of the estimated integrated upstream bottleneck rates of this flow at its alternative outgoing interfaces at the current node.

It is worth mentioning that the availability of each interface may vary based on network conditions. Thus, we require that a fraction of Interest traffic be randomly forwarded to probe potential paths. The ratio of randomly forwarded Interest traffic in our strategy is set to 1% after extensive testing, since this is optimal for most scenarios.

At first glance, the proposed strategy appears straightforward, yet it enables effective load balancing in an efficient and low overhead manner. The experimental results in Section V confirm this as well. Obviously, further potential improvements will be investigated in the future.

F. Consumer Rate Adjustment

In QSCCP, the Consumer Rate Adjustment (CRA) module performs a rate-based adjustment mechanism at the consumer side. Specifically, when a Data packet arrives at the consumer, the CRA module retrieves the suggested target rate of this flow from the *TargetRate* field in the received Data packet. Given that this feedback rate is the estimated suggested rate of Data packets that the consumer should receive, the CRA module must deflate it to determine the suggested Interest sending rate. Thus, the evolution of the Interest sending rate of a flow is defined by:

$$R_I(t) = \beta \cdot R_I(t') + (1-\beta) \cdot \frac{R(t)}{S_D}. \tag{13}$$

Here, $R_I(t')$ represents the Interest sending rate (in packets per second) before the Data packet reaches, $R(t)$ is the carried rate (in bits per second) inside the received Data packet, β is the smooth factor which prevents frequent rate oscillations, and S_D denotes the size (in bits) of the received Data packet.

Using the proposed mechanism, the consumer can rapidly increase or decrease the Interest sending rate to reach a suitable value regardless of whether the network is overloaded or idle. Compared to conventional window-based policies, this drastically reduces the convergence time necessary to reach the steady-state throughput.

Note that the result of equation (13) is primarily a recommended Interest sending rate. The actual Interest sending rate of a flow should not exceed this value but could be lower for applications with modest resource demands, such as voice flows, which typically require merely tens of kilobytes of bandwidth. In addition, for a multipath flow, the Interest sending rate determined by equation (13) at the consumer side is the aggregate available rate for all transmission paths to the content source(s). The specific traffic distribution on each path is influenced by collaboration between the PECN module and AF module, as described in Section IV-D and Section IV-E.

In addition, the value of data size S_D is known a priori in many applications [42], such as sensor readings for IoT applications, video streaming, and Internet telephony applications. In situations where S_D is not known beforehand, the consumer maintains an expected data size estimate and adjusts it accordingly as Data packets are received.

V. PERFORMANCE EVALUATION

In this section, we conduct extensive experiments to evaluate the performance of QSCCP. Initially, we conduct class-based inter-flow competition experiments to verify the effectiveness of QSCCP in meeting diverse QoS preferences. Then, we compare QSCCP with well-known NDN congestion control protocols. Numerous experimental results demonstrate the outstanding performance of our protocol.

A. Experiment Setup

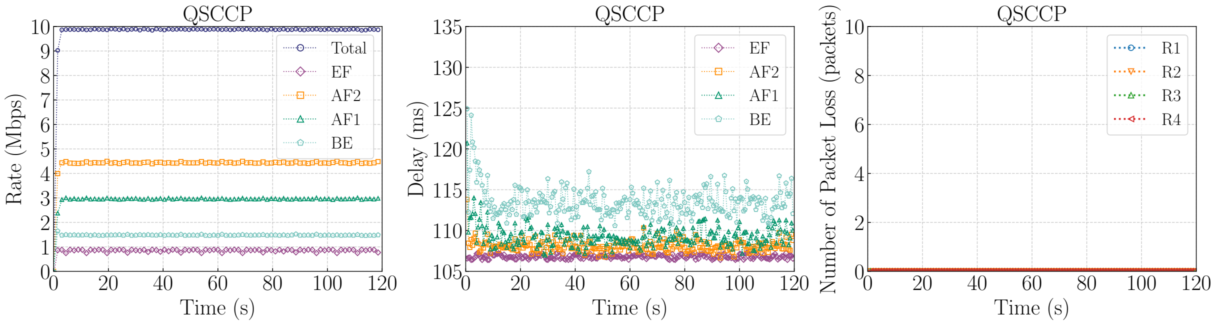
1) Benchmarks Selection: Three well-known NDN congestion control protocols are chosen as benchmarks in our experiments: MIRCC [14], OMCC-RF [13], and PCON [7]. MIRCC is a representative rate-based scheme inspired by the classical IP RCP algorithm, as described in Section II. We select MIRCC as one of the comparison schemes because its rate-driven design motives are comparable to ours. OMCC-RF consists of a per-route round-trip delay estimate-based AIMD policy and a forwarding strategy that relies on Pending Interests. PCON is a state-of-the-art ECN-style scheme that has been integrated into the codebase of the NDN official project. QSCCP is also compared to OMCC-RF and PCON, both of which are representative window-based approaches with multipath forwarding support. The above schemes are evaluated on ndnSIM [43] in NS-3.

2) Parameters Selection: In all experimental scenarios, the payload of a Data packet is set to 1024 bytes following the default setting in ndnSIM. The target link utilization η in equation (1) is set to 0.99 to accommodate abrupt traffic. In order to achieve the tradeoff between responsiveness and stability of QSCCP, we set the smooth factor β in equation (13) to 0.8, and the control parameter γ used in flow state judgment to 0.95, depending on several tests. In addition, the greedy status judgment threshold λ is set to 10 rounds to avoid misjudgments induced by traffic bursts. Moreover, for comparison schemes, all parameter settings in our implementations are identical to those in their original articles.

B. Class-based Inter-flow Competition

1) Scenario 1 (dumbbell topology): In this scenario, we evaluate the performance of QSCCP using a dumbbell topology shown in Fig. 8, where consumers with different service classes compete for a shared bottleneck bandwidth. Each of the four service classes (EF, AF2, AF1, and BE) corresponds to 10 consumer applications. The limited total available bandwidth for delay-sensitive traffic with the service class EF is configured to 1Mbps. The middle level scheduling weights of delay-tolerant traffic with service classes AF2, AF1, and BE adhere to the formula $\omega_{\mathrm{AF2}} : \omega_{\mathrm{AF1}} : \omega_{\mathrm{BE}} = 3{:}2{:}1$. The bottom level scheduling weight of each flow is identical to that of other flows within the same service class. The capacity of the bottleneck link (R2-R3) in the network is 10Mbps. The propagation latency of each consumer-producer path is 100ms. The experiment lasts 120s.

Fig. 7(a) shows the instantaneous throughput of QSCCP in scenario 1. The aggregate throughputs of 10 consumers with service class EF are approximately 1Mbps, which conforms to its limited bandwidth in this scenario. The aggregate throughputs of 10 consumers with other service classes (AF2, AF1, and BE) are around 4.5Mbps, 3Mbps, and 1.5Mbps, respectively, which are proportional to their weight ratio of 3:2:1. These results demonstrate that QSCCP can rationally allocate bandwidth resources depending on the unique requirements of consumers with diverse service classes. We also notice that QSCCP achieves near-perfect bandwidth utilization, while its throughput curves swiftly converge to the corresponding steady state at the beginning of the experiment and thereafter remain stable. This is made possible by the explicit rate feedback and rate-based consumer adjustment mechanism in QSCCP. The instantaneous delay of QSCCP in scenario 1 is illustrated in Fig. 7(b). It is observed that for traffic with service class EF, which has a strict delay requirement, its round-trip delay is close to the propagation latency and the jitter is very small. For traffic with service classes AF2 and AF1, which may tolerate a certain delay, their round-trip delay curves are similar, fluctuating within a small and tolerable range and barely exceeding the propagation latency. The round-trip delay for traffic with service class BE, which has no delay requirements, tends to fluctuate within an acceptable range. These results demonstrate further that QSCCP can provide differentiated delay guarantees to consumers with varying delay requirements. Additionally, as



(a) Throughput (b) Delay (c) Packet Loss

Fig. 7. Instantaneous transmission performance of QSCCP in scenario 1.

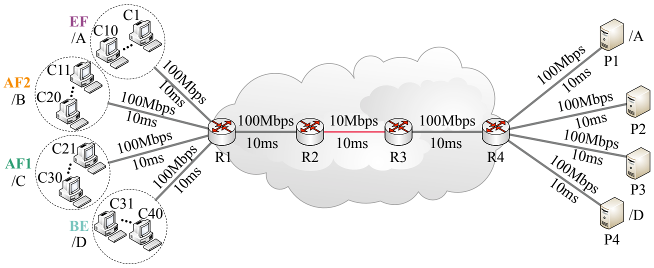


Fig. 8. Dumbbell topology for scenario 1.

illustrated in Fig. 7(c), packet loss and retransmissions are not observed in this scenario because of the precise rate-based feedback mechanism in QSCCP.

2) Scenario 2 (tree-structured topology): In this scenario, we examine the dynamics of QSCCP using a typical tree-structured topology shown in Fig. 10, in which flows dynamically join or exit the network. Consumer C1 with service class AF2 sends requests from 0s to 90s. Consumer C2 with service class AF1 sends requests from 30s to 120s. Consumer C3 with service class BE sends requests from 60s to 150s. The weight ratio of these three service classes (AF2, AF1, and BE) remains unchanged; i.e., at 3:2:1. Each consumer has two alternative transmission paths to obtain the requested contents. The propagation latency of each consumer-producer path is 60ms. The experiment is 150 seconds long.

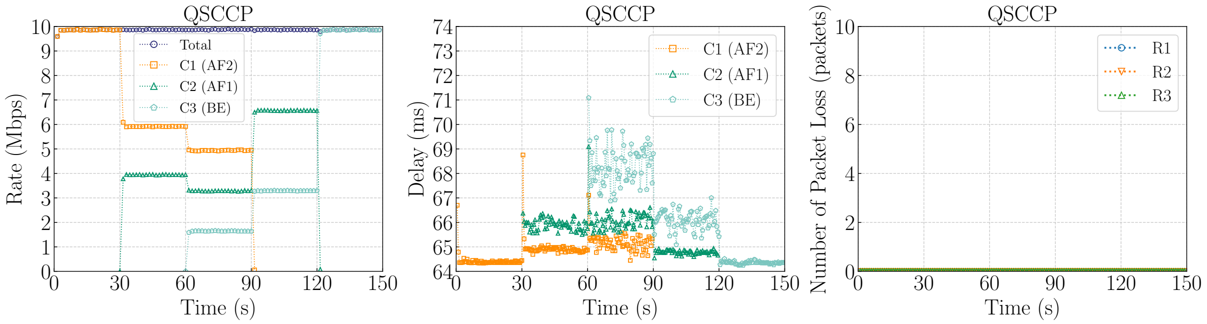
Fig. 9(a) shows the instantaneous throughput of QSCCP in scenario 2. During the intervals 0s to 30s and 120s to 150s, the entire bandwidth is occupied by a single active consumer. Multiple active consumers share the bottleneck bandwidth (the link R1-R2 and the link R1-R3) during other periods, and their throughputs are proportional to their individual bandwidth demands (corresponding to the service class). We further observe that whenever a flow joins or exits, the active consumer(s) will rapidly adjust the Interest sending rate(s) to attain the new throughput equilibrium(s) in a short period of time, which is commensurate with the respective service class(es) and maintains maximum network utilization. This excellent responsiveness is a result of the PECN module in QSCCP, which estimates and notifies consumers of the latest available network resources using only a few round-trip Interest packets and Data packets. Fig. 9(b) shows the instantaneous delay of QSCCP in scenario 2. As anticipated, the round-trip delay of each consumer during each period is consistent with the delay constraint imposed by the associated service class. Additionally, the amount of packet loss in this scenario (Fig. 9(c)) remains at zero, the same as scenario 1. The above results further verify the excellent QoS-aware ability and transmission performance of QSCCP in a tree-structured topology.

C. Multipath Transmission with Heterogeneous Bandwidths / Latencies

1) Scenario 3 (diamond-structured topology): As shown in Fig. 11, using a diamond topology, we compare the multipath transmission performance of QSCCP and three other well-known NDN congestion control protocols. In practice, diamond topology is the most prevalent topology in which network traffic on different paths is split and aggregated at certain nodes [15]. As the existing NDN congestion control solutions, including the three selected classic comparison schemes, have no effective QoS considerations, the comparative experiments mainly focus on crucial transmission performance metrics, such as throughput, fairness, delay, and packet loss. We conduct two sets of experiments: Case I, which uses heterogeneous bandwidths and homogeneous latencies, and Case II, which uses homogeneous bandwidths and heterogeneous latencies [16]. In both cases, three consumers (C1, C2, and C3) with the same service class send content requests with different name prefixes at intervals of 80 seconds each. However, around 240s, consumer C3 abruptly changes his behavior to greedy by developing a high and constant Interest sending rate, which in this setting amounts to 1160 Interest packets per second (the corresponding theoretical Data returning rate is approximately 10Mbps). Additionally, in both cases, the total bottleneck bandwidth within the diamond is 14Mbps. The experiment runs for 320 seconds.

a) ***Throughput:*** The instantaneous throughput of each scheme is displayed in Fig. 12. It is important to discuss the experiment results in both cases before the consumer C3 grows greedy (0 to 240s), and then discuss the results afterward (240 to 320s).

From 0 to 240 seconds, with the exception of PCON, all schemes achieve approximative multi-flow fairness, whereas



(a) Throughput (b) Delay (c) Packet Loss

Fig. 9. Instantaneous transmission performance of QSCCP in scenario 2.

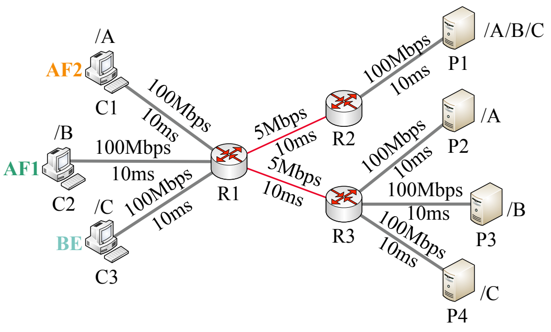


Fig. 10. Tree-structured topology for scenario 2.

QSCCP outperforms other schemes in terms of convergence rate, stability and link utilization. QSCCP exhibits identically excellent performance regardless of the heterogeneous bandwidth or latency experienced during multipath transmission. Specifically, in both cases, QSCCP adjusts the Interest sending rate to the new steady state in a short time whenever a new flow joins the network (i.e., 0s, 80s, and 160s), for the same reasons as in the first two scenarios. It also achieves fair bandwidth resource allocation among active consumers with the same resource demand. At each steady stage of QSCCP, its instantaneous throughput curves are always smooth compared to other schemes. This outstanding performance can be attributed to the reasonable flow rate estimation and feedback provided by the PECN module, which plays a crucial role in such diamond topologies with many branch nodes. Due to the additional benefit of the AF module, which dynamically adjusts forwarding probability in response to feedback rate information, QSCCP has the best link utilization. MIRCC has a great convergence rate and a gently smooth steady state in both cases. This is actually the common advantage of rate-based consumer policies, which can derive the appropriate Interest sending rate explicitly through the returned Data packets carrying feedback rate rather than implicitly through window probing. In contrast to QSCCP, the steady state of MIRCC exhibits minor oscillations. The reason is that compared to QSCCP, the rate estimation in MIRCC is more sensitive to the variation in queue length during multi-flow competition on bottleneck links. The total throughput of MIRCC is a fraction lower than that in QSCCP. In the original design of MIRCC, the link utilization parameter is set at 0.95 to avoid oscillations induced by packet queuing or congestion overshoot. As for OMCC-RF and PCON, their throughput curves are saw-toothed in both cases owing to their AIMD-style window policies at consumers, which implicitly probe available network resources. OMCC-RF uses the number of Pending Interests (PI) as the congestion metric of each interface, despite the fact that PI does not accurately describe the congestion status of a single path. Consequently, in Case I, the path with the lower bandwidth is overloaded first, triggering the premature window decrease at consumers, while other paths with larger bandwidth are underused. In Case II, the path with the lower delay becomes congested earlier, prompting consumers to reduce the sending rate. In both cases, the low link utilization for OMCC-RF is caused by these unjustified window reductions, especially in Case I. For PCON, it exhibits the most severe throughput oscillations with the slowest convergence rate, due to the lack of coordination between its forwarding strategy and BIC-based window policy. As described in Section II, its forwarding strategy dynamically shifts network traffic away from congested paths based on congestion marks received at each interface. However, these congestion marks also lead to window reductions for consumers. These recurrent controls inevitably result in throughput oscillations and slow convergence. Worse yet, in such a complex diamond topology, congestion marks are aggregated and dispersed across multiple paths. Therefore, downstream routers in PCON are unable to accurately identify the actual links where congestion occurs. This exacerbates the oscillations and unfairness during multi-flow competition, especially in Case I with heterogeneous bandwidths.

From 240s to 320s, only QSCCP swiftly identifies the misbehaving consumer C3 and restrains its greedy behavior in both cases. In contrast, the other three schemes have their hands tied, as much of the available bandwidth is unseasonably occupied by the greedy flow (C3), while normal consumers (C1 and C2) endure poor network performance. This is because QSCCP offers a unique punishment tactic for greedy flows that the other three schemes fail to consider. In QSCCP, intermediate nodes periodically detect the states of passing flows, and punish greedy flows by limiting their packet scheduling with the discount factor φ (0.5 by default). The remaining available link capacity is shared proportionally among

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

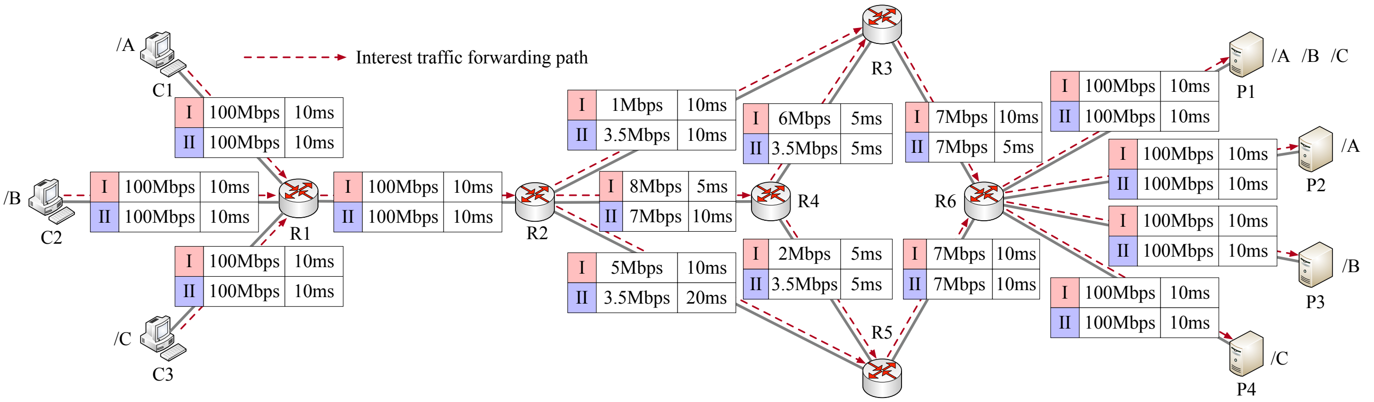


Fig. 11. Diamond-structured topology for scenario 3.

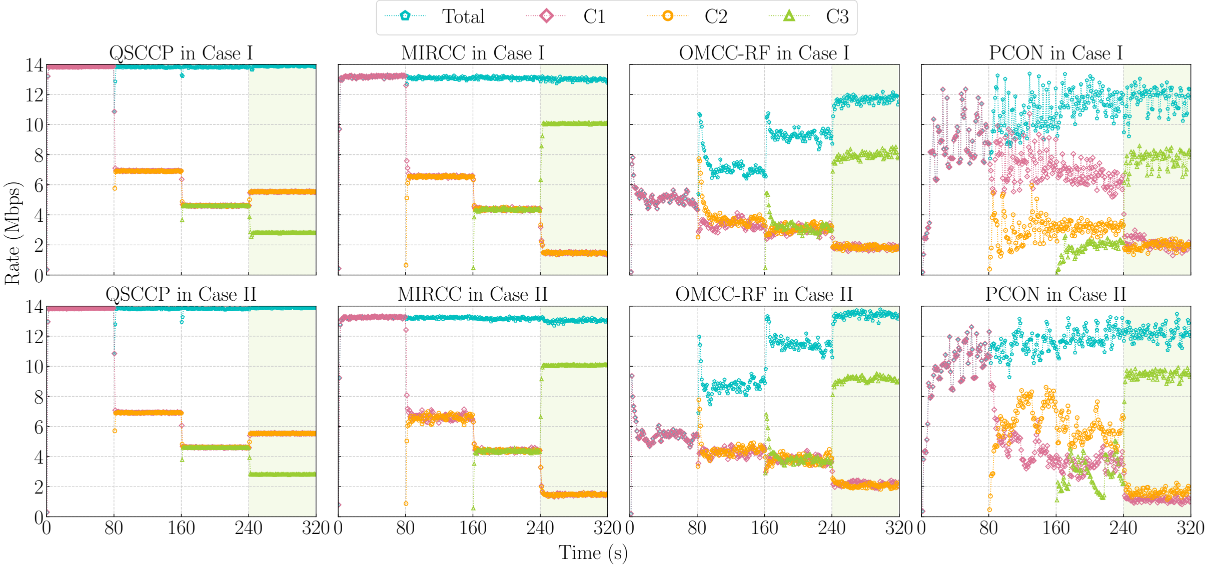


Fig. 12. Instantaneous throughputs of four schemes in scenario 3 (The green shaded areas represent the greedy phase of C3).

normal flows based on their respective weights, as described in Sections IV-C and IV-D. Thus, the actual throughput ratio of three consumers (C1, C2, and C3) during the punishment period (240s to 320s) for QSCCP in this scenario is around 2:2:1.

*b) **Delay**:* In Case I, the transmission paths have homogeneous latencies and the theoretical round-trip propagation latencies are all 100ms. Instead, in Case II, the transmission paths have heterogeneous latencies, with theoretical round-trip propagation latencies of 90, 100, 110, or 120ms. Fig. 13 presents the instantaneous delay of each scheme in both cases. From 0 to 240 seconds, QSCCP has a lower and more steady delay than MIRCC and PCON, which is a result of its packet scheduling design and fast response ability, as previously described. Meanwhile, QSCCP is not susceptible to heterogeneous path bandwidth or latency settings, like its throughput performance. Such remarkable performance is critical for ubiquitous multimedia services over networks. After 240s, the delay of the greedy flow C3 increases, as its packet forwarding is limited by the punishment tactic in QSCCP, and normal flows are protected from C3's interference. In contrast to its steady throughput performance, MIRCC's delay oscillates dramatically in both cases. To strike a balance between inter-flow fairness and network utilization, MIRCC adopts a dual-class best-subflow scheme. However, secondary traffic always yields to primary traffic, and full queues are invariably emptied in a radical way. This fuels the drastic instability for delay in such a multi-flow and multipath competition scenario. OMCC-RF has a comparatively low delay due to the low link utilization in both cases. In Case I, the delay fluctuates with peaks because of the large queuing delays on low bandwidth paths which become congested earlier. The delay fluctuations become more intense when the consumer C3 gets greedy. In Case II, the low delay paths are filled first and trigger the window decrease at the consumer, whilst the higher latency paths are underused and have lower queuing delays. Consequently, the delay fluctuations of OMCC-RF in Case II fall within a reasonable range. Regarding PCON, delay results have many peaks in Case I, but they fluctuate routinely in Case II. The reason for this is that when the bandwidth discrepancies between paths are quite substantial (Case I), the oscillation of its forwarding strategy will be more aggressive.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

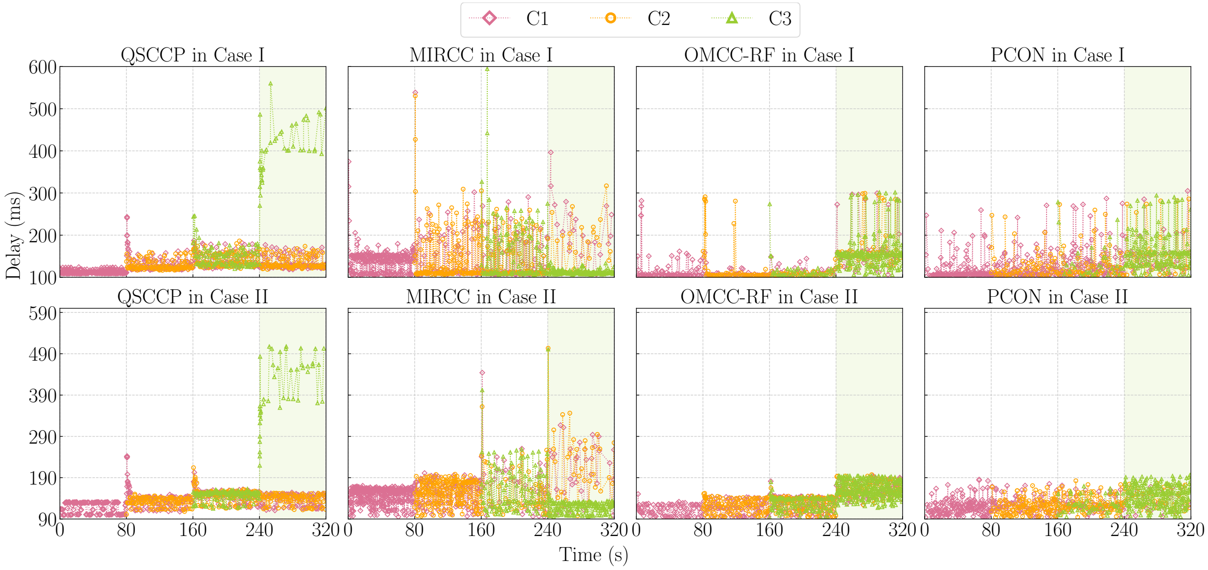


Fig. 13. Instantaneous delays of four schemes in scenario 3 (The green shaded areas represent the greedy phase of C3).

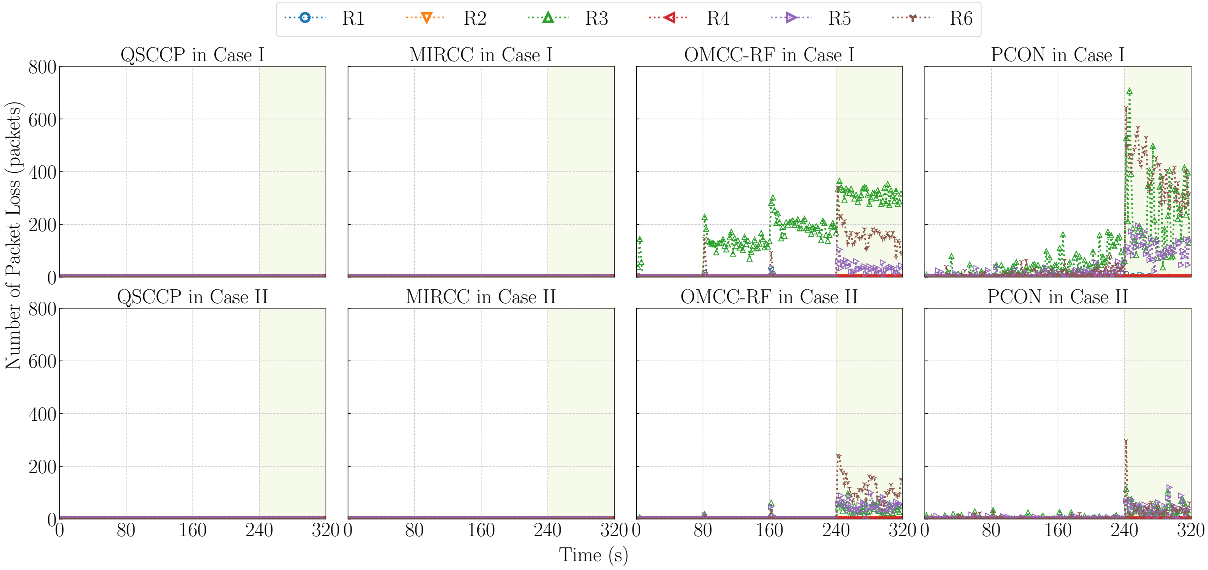


Fig. 14. Instantaneous numbers of packet loss of four schemes in scenario 3 (The green shaded areas represent the greedy phase of C3).

This results in larger queuing delay oscillations on paths with lower bandwidth.

c) ***Packet loss**:* Fig. 14 illustrates the instantaneous numbers of packet loss of four schemes. Note that "packet loss" refers to the Data packets dropped at each intermediate router. As noted previously, the results demonstrate that QSCCP can prevent packet loss in both cases due to its rational resource allocation and outstanding performance against the greedy flow. MIRCC also has zero dropped packets because of its accurate rate estimation. In addition, its Interest shaper deployed on each forwarder can prevent packet loss caused by the greedy behavior of consumer C3, but it cannot restrain the unfair bandwidth preemption induced by this greedy flow, as shown in Fig. 12. In contrast to QSCCP and MIRCC, OMCC-RF suffers from a large number of dropped packets due to its inaccurate PI-based congestion metric and inefficient AIMD-driven window policy, notably in Case I. In Case II, when traffic is distributed evenly, packet loss is reduced. Similarly, PCON's BIC-based window policy will necessarily lead to some packet drops during multi-flow competition, while its forwarding strategy is more hostile in a heterogeneous bandwidth environment, resulting in more dropped packets in Case I. After 240 seconds, the performances of OMCC-RF and PCON deteriorate in both cases because neither can handle the greedy flow of C3.

D. Impact of In-Network Caching

1) Scenario 4 (impact of in-network caching): In this scenario, we evaluate the impact of in-network caching on the proposed congestion control solution. In the topology shown in Fig. 15, two consumers (C1 and C2) are set to retrieve the same data, with C2 starting 5 seconds later than C1.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

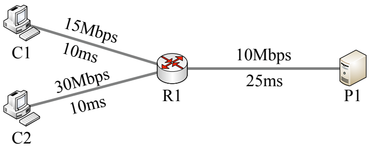


Fig. 15. Topology for scenario 4.

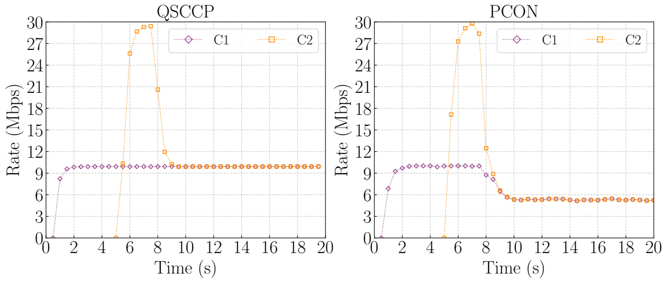


Fig. 16. Instantaneous throughputs of QSCCP and PCON in scenario 4.

We simulate a situation where the consumers can access a nearby high-bandwidth in-network cache and a remote lower-bandwidth content repository. Router R1 enables in-network caching, employing a FIFO policy, and the size of the content store is set to 10,000 packets. The capacity of the bottleneck link (R1-P1) is set to 10 Mbps.

Fig. 16 shows the evaluation results. C1 starts first and retrieves data from P1, adjusting its Interest sending rate to adapt to the bottleneck bandwidth of 10Mbps on the R1-P1 link. Its data is admitted to the in-network cache of R1. At 5 seconds, C2 starts and is initially satisfied by the cache at R1. The high bandwidth between C2 and R1 allows C2 to quickly catch up with C1. When C2 exhausts the cache at around 8 seconds, it joins C1 in retrieving data from P1. Since QSCCP's congestion control does not rely on RTT measurements as a congestion indicator but instead uses precise feedback information, its throughput is not affected by potential RTT oscillations caused by the cache hit. In the steady state, in QSCCP, the throughput of C1 and C2 steadily converges to 10Mbps. PCON, by setting a higher fixed RTO and using explicit marks for window adaptation, achieves fast congestion reaction and thus also avoids similar issues associated with RTT oscillations. Compared to the precise feedback information provided by the PECN module of QSCCP, PCON's simple explicit marks indicate only congested or uncongested states. This causes PCON to react more slowly when the cache is exhausted, generating more congestion marks. Both consumers of PCON reduce their windows upon receiving congestion marks. Due to PCON's BIC-based window adjustment, which becomes more conservative after this window reduction, in the subsequent steady state, PCON's throughput converges to around 6Mbps, lower than that of QSCCP. Neither QSCCP nor PCON caused any packet loss or retransmissions during this process.

E. Multiaccess Network

1) Scenario 5 (Multiaccess network): In this scenario, we validate the effectiveness of the proposed scheme in a multiaccess network environment. We simulate a scenario as shown in Fig. 17, where three edge devices, N1, N2, and N3, are connected to the network using multiple access technologies. Specifically, edge device N1 is connected to the network using two access technologies, including Fast Ethernet and Wi-Fi (IEEE 802.11a). N2 and N3 are connected to the network using Ethernet. Each edge device both consumes and produces data simultaneously.

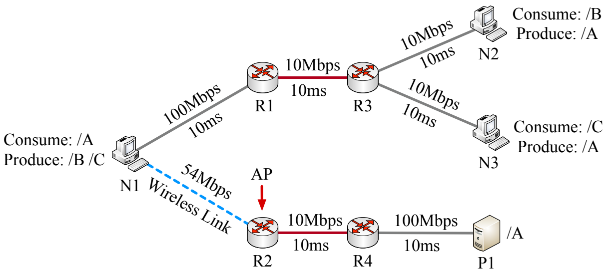


Fig. 17. Topology for scenario 5.

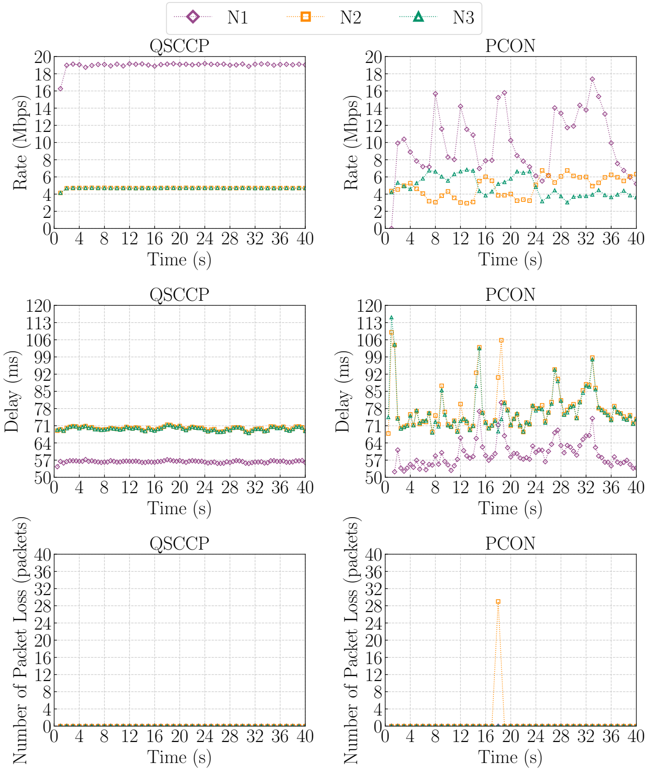


Fig. 18. Instantaneous transmission performance in scenario 5.

Fig. 18 shows the instantaneous transmission performance of QSCCP and PCON in the multiaccess network scenario. As can be seen, QSCCP significantly outperforms PCON in terms of throughput, latency, and packet loss. For QSCCP, due to the feedback information provided by the PECN module carried in the returned Data packets, the edge devices connected through different access technologies in the network can quickly adjust to an appropriate Interest sending rate. This leads to faster convergence, higher link utilization, and superior steady-state performance compared to PCON's window adjustment based on simple congestion marks. In contrast to the severe fluctua-

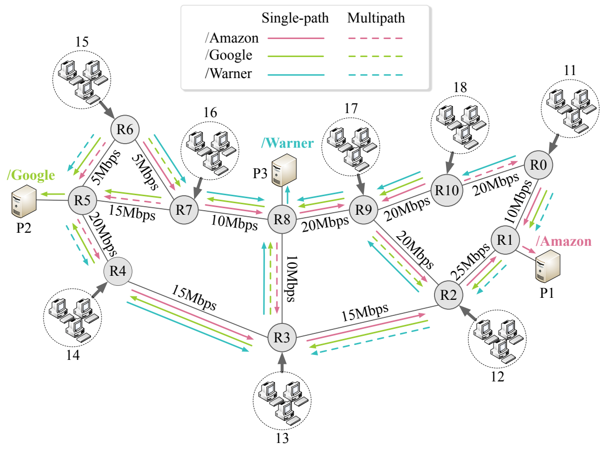


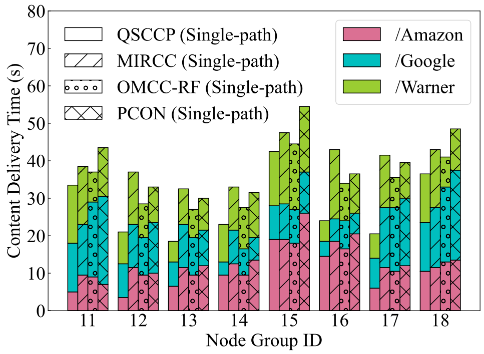
Fig. 19. Abilene topology for scenario 6.

tions observed with PCON, QSCCP exhibits greater stability and lower latency. These excellent results of QSCCP validate its effectiveness for multiaccess networks.

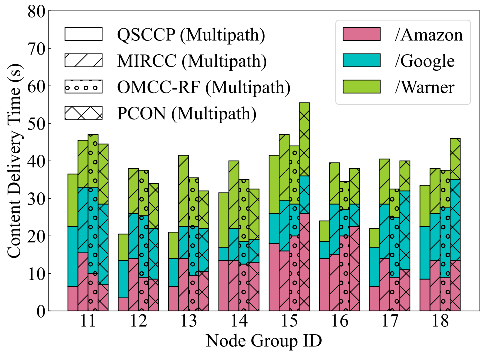
F. Content Delivery in Complex Topologies

1) Scenario 6 (Abilene topology): In this scenario, we consider the content delivery performance of our proposed scheme and comparison schemes, which is one of the most critical performance metrics for Internet users. As depicted in Fig. 19, we use a real backbone network topology named Abilene topology. There are three producers (P1, P2, and P3), and each of them stores the content under a given name prefix (/Amazon, /Google, or /Warner). Each content consists of 5000 chunks, each of which is 1024 bytes in size. Each dotted circle indicates a node group with three consumers, each of whom requests content from one of the three producers. The delay of each link is set at 10ms. The routing configuration is identical to the setup in [13], which can enable or disable multipath forwarding.

Fig. 20 illustrates the content delivery times of four schemes per name prefix and per node group in two cases: single-path forwarding and multipath forwarding. At each node group, regardless of whether multipath is enabled, QSCCP manifests the shortest content delivery time among the four solutions due to its notable convergence rate. When multipath forwarding is enabled, the PECN module of QSCCP effectively perceives the traffic variation in upstream and downstream paths, providing instant accurate feedback in this hybrid topology. The AF module of QSCCP is essential for balancing traffic load and improving link utilization. These outstanding features of QSCCP create reasonable resource allocation among flows in such a realistic cross-traffic scenario and reduce the additional delay caused by retransmissions. For MIRCC, its dual-class best-subflow scheme handles cross-traffic poorly, resulting in a considerably longer content delivery time. And its path management scheme deployed at consumers cannot respond flexibly to network load variations when there are multiple active bidirectional flows. Moreover, MIRCC assumes that the entire reverse link bandwidth is available for returned Data packets when calculating the flow rate, resulting in biased rates. Instead, QSCCP takes into account the bandwidth expensed by Interest packets in the reverse direction of the link. OMCC-RF has a faster content delivery time than MIRCC and PCON. According to our experiments, this time gain comes at the expense of considerable packet loss. As expected, PCON's oscillating forwarding strategy and repetitive control reduce performance in this hybrid cross-traffic contention environment.



(a) Single-path



(b) Multipath

Fig. 20. Content delivery times of four schemes in scenario 6.

2) Scenario 7 (GEANT topology): In this scenario, we compare QSCCP with other schemes under a larger real network topology, namely GEANT topology [44], as shown in Fig. 21. GEANT is a European academic network that connects to research networks in many countries worldwide. The GEANT topology's core network consists of 46 nodes. In our experiment, nodes 1, 2, 7, 8, 19, 27, 38, and 44 are selected to attach content repositories, each providing the content under a given name prefix (/A, /B, /C, /D, /E, /F, /G, /H). Besides, we attach eight consumers to each of the nodes 3, 4, 18, 25, 37, 40, 41, and 45. Each of the eight consumers requests content from one of eight content repositories. Each content consists of 5000 chunks, each of which is 1024 bytes in size. The delay of each link is set to 10ms.

Fig. 22 displays the content delivery times of the four schemes per name prefix and per node group. QSCCP consistently outperforms competing schemes at all nodes. This further validates the feasibility of deploying QSCCP in a realistic large-scale topology with substantial cross-traffic. QSCCP can improve the overall QoE for Internet users by providing instant service assurance, rapid content delivery, and

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

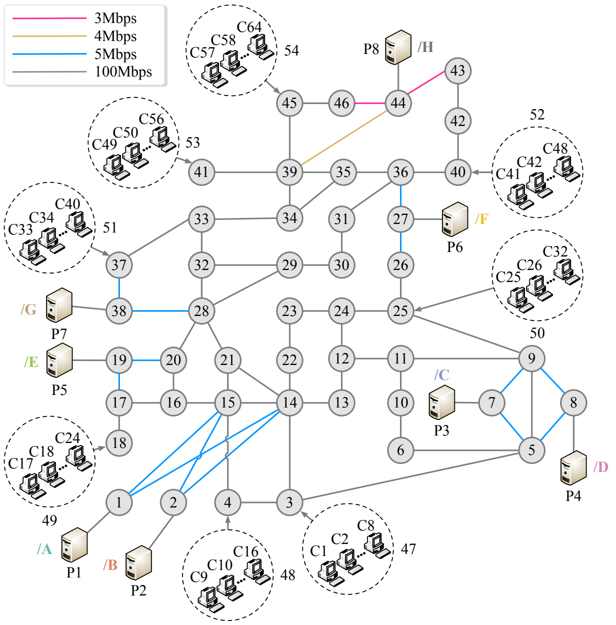


Fig. 21. Extended GEANT topology for scenario 7.

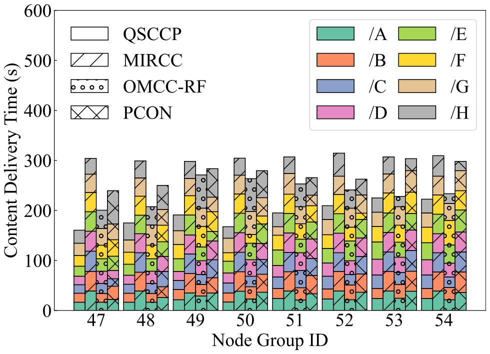


Fig. 22. Content delivery times of four schemes in scenario 7.

stable transmission performance.

G. Overhead Analysis

In our implementation, as detailed in Section IV-B, the Interest packet has two new fields: *ServiceClass* (1-byte unsigned integer value) and *DownstreamRate* (8-byte unsigned integer value). The Data packet also has two new fields: *ServiceClass* (1-byte unsigned integer value) and *TargetRate* (8-byte unsigned integer value). Each packet is encoded via a general TLV (Type-Label-Value) scheme [1]. The original sizes of an encoded Interest packet and an encoded Data packet are 21 bytes and 1052 bytes, respectively. With the extra fields in QSCCP, the packet sizes increase to 34 bytes and 1065 bytes, respectively. This results in an approximately 2.4 percent of additional packet overhead during a round-trip transmission. Besides, the two introduced hash tables (i.e., DRT and TRT), as described in Section IV-D, have a minimal lookup overhead with a time complexity of O(1). The optimization of their storage space has been earmarked as future work. In a word, the overall overhead of QSCCP is both reasonable and allowable considering the significant performance gains discussed earlier.

VI. CONCLUSION

In this paper, we present a QoS-aware congestion control protocol (QSCCP) built on NDN, a mainstream ICN paradigm, to enhance content delivery efficiency as well as satisfy diverse QoS preferences for various applications. The implementation details of QSCCP mainly comprise CDS, PECN, AF, and CRA modules. Specifically, the CDS module performs multi-level, class-based differentiated scheduling for various content requests. The PECN module provides and notifies precise feedback information about available network resources and optimizes resource allocation, especially for multipath flows. The AF module controls the forwarding probability of each interface at routers in cooperation with PECN. The CRA module adjusts the Interest sending rate based on the feedback information carried within Data packets. Extensive experimental results demonstrate the effectiveness and superiority of QSCCP over the state-of-the-art NDN congestion control protocols in terms of throughput, fairness, delay, and packet loss, with outstanding QoS guarantees, stability, and convergence rate.

Our ongoing and future initiatives are centered on four key aspects. Firstly, to delve into the optimal design of congestion control algorithms for ICN networks, it is crucial to build a mathematical model for accurately analyzing and predicting network behavior. The recent quantitative theory of bottleneck structures (QTBS) [40], [41] provides a general mathematical framework for modeling communication networks. It can be used as a rigorous formal framework to evaluate the performance of our scheme and offer a benchmark against an optimal baseline. In our subsequent endeavors, we plan to model the behavior of QSCCP based on the QTBS model and extend it to explore the optimal design of network transport protocols under the unique traffic patterns of NDN. Meanwhile, we will leverage the predictive capabilities of the QTBS theory, as well as its understanding of congestion control, to further optimize the designs of packet forwarding and flow scheduling.

Secondly, the deployment of ICN infrastructure in current IP-based networks is progressing incrementally. We intend to integrate the recent Software Defined Networking (SDN) architecture [45] into our work. This integration will facilitate the flexible application and extension of the proposed protocol in real-world network environments. Meanwhile, it will provide strengthened forwarding and efficient resource management.

Thirdly, the rapid development of various communication technologies has fostered the widespread presence of multiaccess network scenarios in modern networks. Therefore, it is necessary to fully consider the enhancement of transmission performance in various multiaccess network environments in the NDN architecture. We plan to further explore in detail the application of QSCCP in multiaccess scenarios and pursue additional optimizations.

Lastly, for any congestion control algorithm, conducting large-scale performance testing and further optimization in

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

real-world production networks is an indispensable step for its practical implementation. In this regard, we are currently undertaking relevant experiments on the actual testbed.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to the editors and anonymous reviewers for their constructive suggestions, which have greatly contributed to enhancing the quality of this paper.

APPENDIX A
SUPPLEMENTARY EXPERIMENTS

To further evaluate the feasibility of QSCCP in a network environment that closely resembles real-world conditions, we conduct supplementary experiments using Mininet [46]. The Mini-NDN tool [47] is utilized to construct the NDN network protocol stack within Mininet. It integrates with the NDN libraries, NFD, NLSR, and NDN-tools released by the NDN project. The experimental scenarios and results are detailed in the following subsections.

A. Consistent High-Performance in Identical Scenarios

We select several representative experimental scenarios introduced in Section V to verify the performance of QSCCP on Mininet. The selected scenarios include: *scenario 1* (dumbbell topology, Fig. 8) described in Section V-B, *scenario 2* (tree-structured topology, Fig. 10) described in Section V-B, and *scenario 3* (diamond-structured topology, Fig. 11) described in Section V-C. For each scenario, the configurations remain consistent with their respective original setups. Ten trials are conducted for each scenario, and the final average performance is reported.

The instantaneous transmission performances of QSCCP in these three identical scenarios (note: scenario 3 consists of two cases) are respectively depicted in Fig. 23, Fig. 24, Fig. 25, and Fig. 26. We observe that QSCCP consistently maintains superior performance, as discussed in Section V, across all scenarios reproduced in a realistic network environment based on Mininet. It is noticed that the delay of QSCCP experiences a slight increase compared to the simulation results based on ndnSIM. It remains within the expected range and fluctuates within acceptable bounds. This results in the impact of a more realistic experimental environment.

Specifically, as shown in Fig. 23 and Fig. 24, as anticipated, QSCCP meets diverse QoS requirements (e.g., bandwidth, latency, packet loss) during class-based inter-flow competition in scenario 1 and scenario 2. In scenario 3, QSCCP has identically excellent transmission performance and fairness regardless of the heterogeneous bandwidth (Fig. 25) or latency (Fig. 26) experienced during multipath transmission. This can be attributed to QSCCP enhancing the control at the receiver based on accurate feedback from the network, which also makes it converge to fairness for all competing flows. In particular, at 240s of scenario 3, as shown by the green shaded areas of Fig. 25 and Fig. 26, when a greedy flow (i.e., C3) appears in the network, the punishment tactic of QSCCP comes into effect. It swiftly suppresses the preemptive behavior of the greedy flow and fairly distributes the remaining available link capacity to other normal flows (i.e., C1 and C2). Additionally, as shown in Fig. 24, Fig. 25, and Fig. 26, QSCCP demonstrates rapid responsiveness and convergence speed, as well as excellent stability when flows dynamically join and exit the network. These results indicate the feasibility of deploying QSCCP in real-world network environments.

For a more detailed analysis of the experimental results in each scenario, readers can refer to the similar discussions presented in Section V.

B. Deployment Friendliness and Fairness

In the real world, network flows compete with each other. We conduct the following experiments to verify the friendliness and fairness of QSCCP under multi-flow competition. We compare QSCCP with four classic congestion control schemes: Reno, BIC, CUBIC, and BBR. These four comparison algorithms are ported and implemented in NDN, and the first three schemes are implemented based on PCON [7].

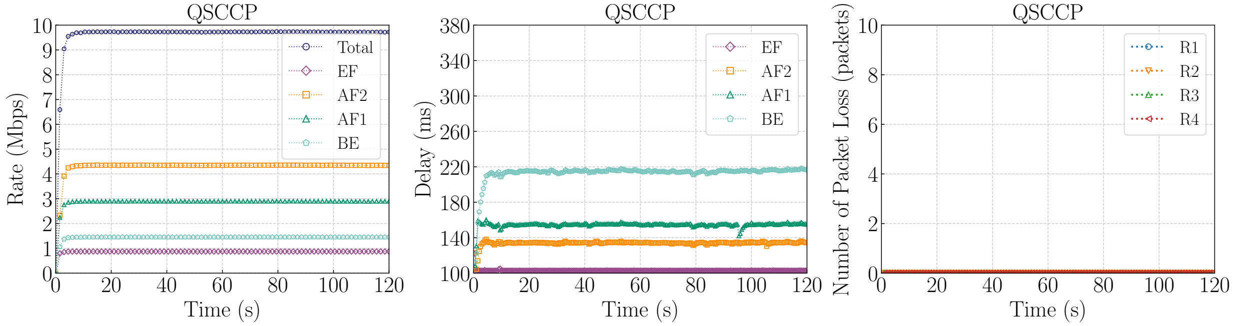
1) Friendliness: First, we assess the friendliness of QSCCP using a dumbbell topology like Fig. 8, where two users compete for a shared bottleneck link. One user is set to use CUBIC, the default congestion control algorithm in most operating systems today. The other user is set to use various test algorithms. We conduct repeated tests for various one-way delays and bandwidths of the bottleneck link, and record the data delivery rate of each flow separately. Each test is run five times.

Fig. 31 shows the average friendliness ratio, defined as the ratio of the delivery rate of the test algorithm to the delivery rate of CUBIC. A higher value of the friendliness ratio indicates that the test algorithm is more aggressive than CUBIC. Fig. 27 and Fig. 28 shows the average link utilization. From Fig. 31, Fig. 27 and Fig. 28, it is evident that QSCCP demonstrates good friendliness compared to other test algorithms when considering CUBIC as a reference. Its performance advantage does not come at the expense of sacrificing the performance of flows using other congestion control algorithms.

2) Fairness: Second, we assess the fairness of QSCCP. We also adopt a similar dumbbell topology like Fig. 8, and two users compete for a shared bottleneck link. Both users use the same test algorithm. Repeated tests are conducted under various one-way delays and bandwidths of the bottleneck link. The data delivery rate of each flow is recorded separately. Each test is run five times.

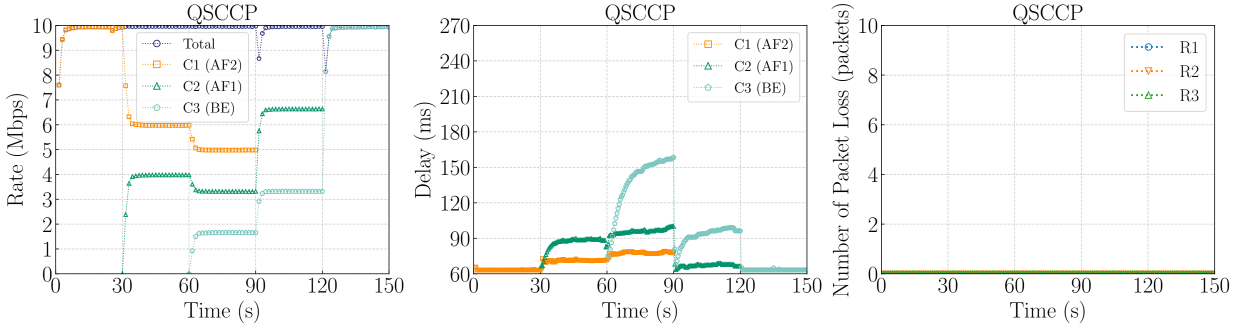
Fig. 32 illustrates Jain's Fairness Index for each scheme under varying bandwidth and latency conditions. A value closer to 1 indicates greater fairness of the test algorithm. Fig. 29 and Fig. 30 shows the average link utilization. From Fig. 32, Fig. 29, and Fig. 30, it can be observed that QSCCP demonstrates near-optimal fairness under various link settings. In contrast, other test algorithms show significant fluctuations in fairness when the link bandwidth or delay changes.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052



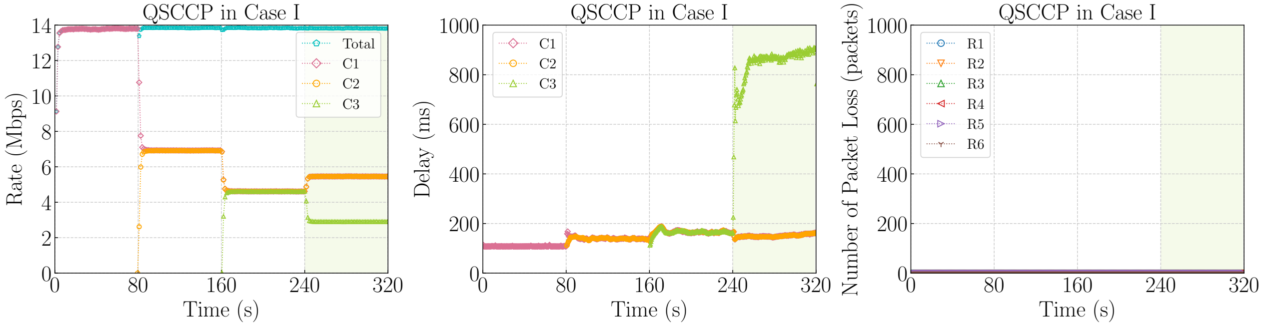
(a) Throughput (b) Delay (c) Packet Loss

Fig. 23. Instantaneous transmission performance of QSCCP in an identical scenario 1.



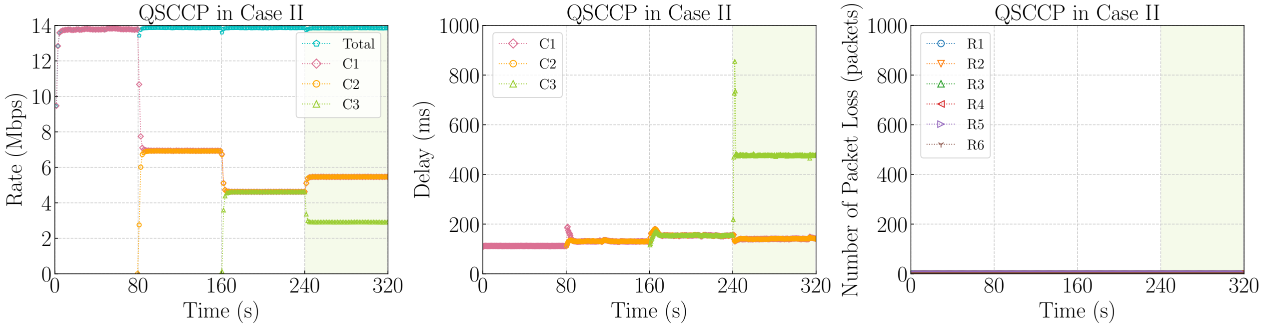
(a) Throughput (b) Delay (c) Packet Loss

Fig. 24. Instantaneous transmission performance of QSCCP in an identical scenario 2.



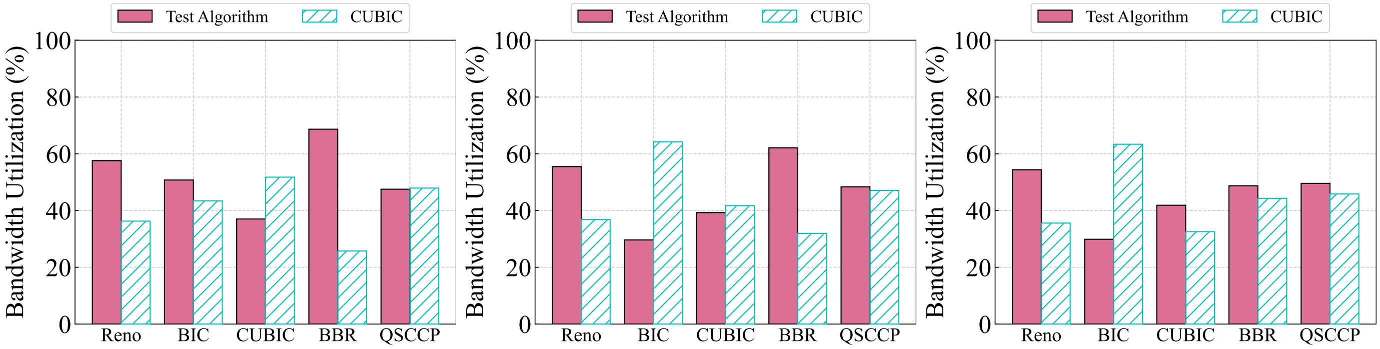
(a) Throughput (b) Delay (c) Packet Loss

Fig. 25. Instantaneous transmission performance of QSCCP in an identical Case I of scenario 3 (The green shaded areas represent the greedy phase of C3).

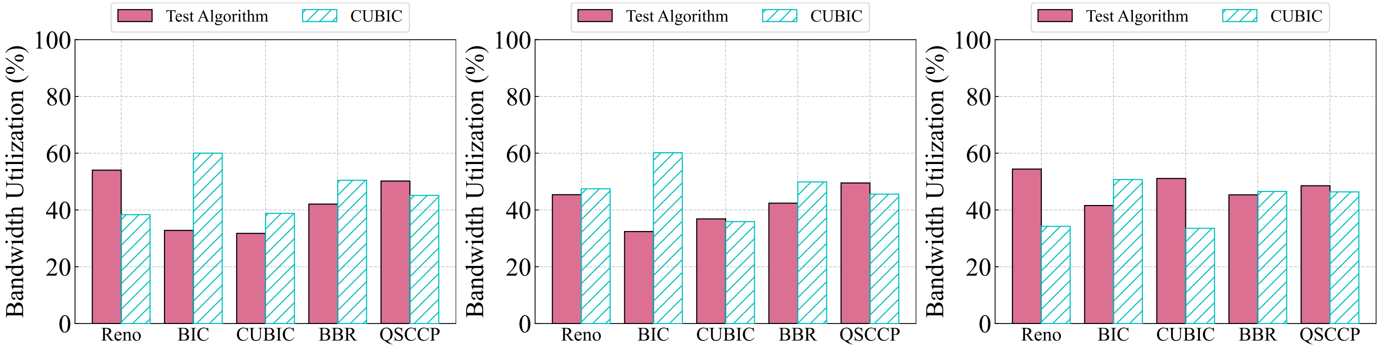


(a) Throughput (b) Delay (c) Packet Loss

Fig. 26. Instantaneous transmission performance of QSCCP in an identical Case II of scenario 3 (The green shaded areas represent the greedy phase of C3).

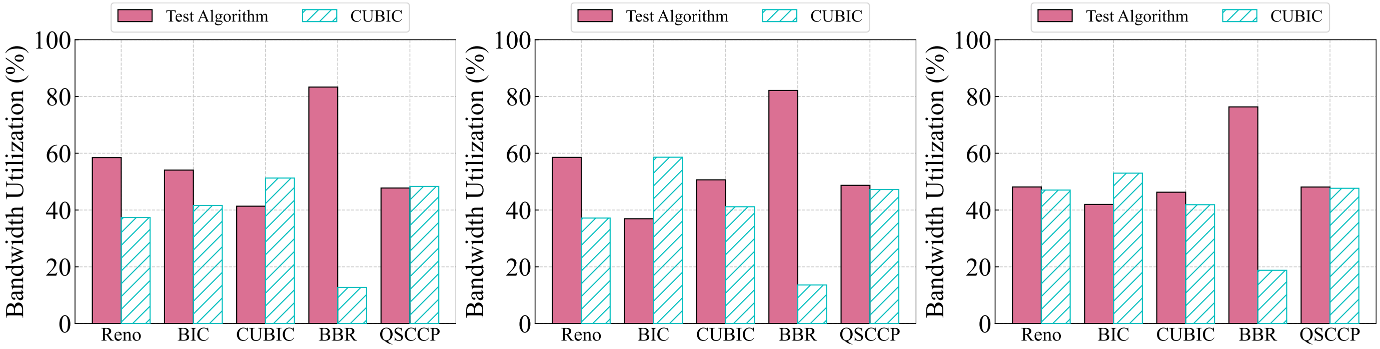


(a) Bottleneck bandwidth of 5Mbps (b) Bottleneck bandwidth of 10Mbps (c) Bottleneck bandwidth of 15Mbps

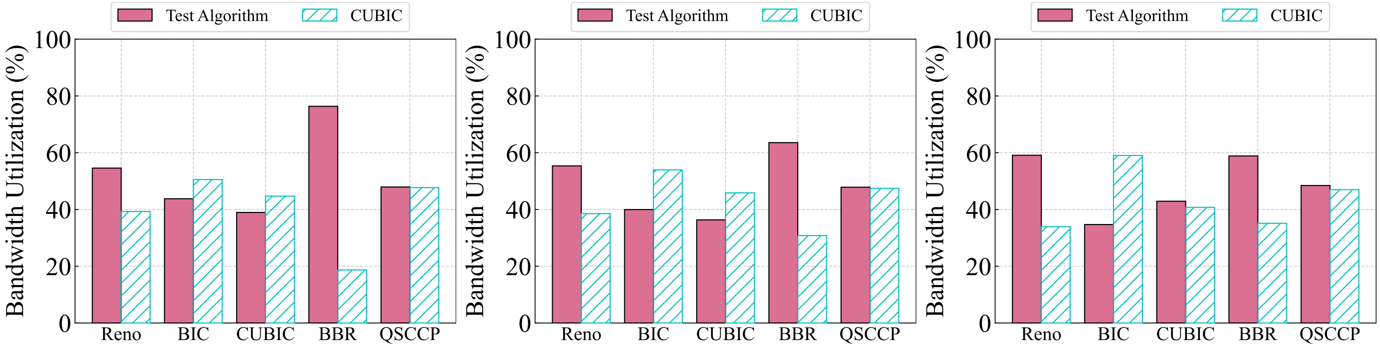


(d) Bottleneck bandwidth of 20Mbps (e) Bottleneck bandwidth of 25Mbps (f) Bottleneck bandwidth of 30Mbps

Fig. 27. Average bandwidth utilization of multiple users during competition with CUBIC under varying bottleneck bandwidth.



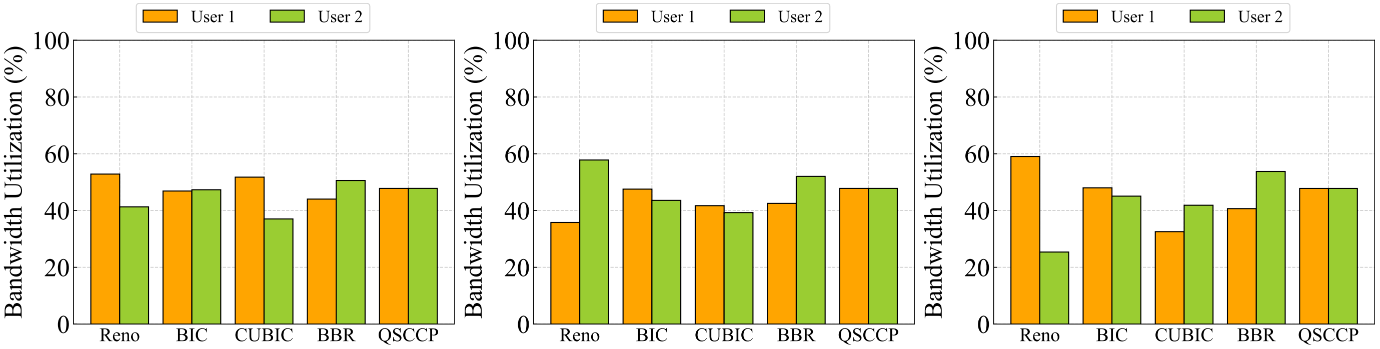
(a) One-way base delay of 10ms (b) One-way base delay of 20ms (c) One-way base delay of 40ms



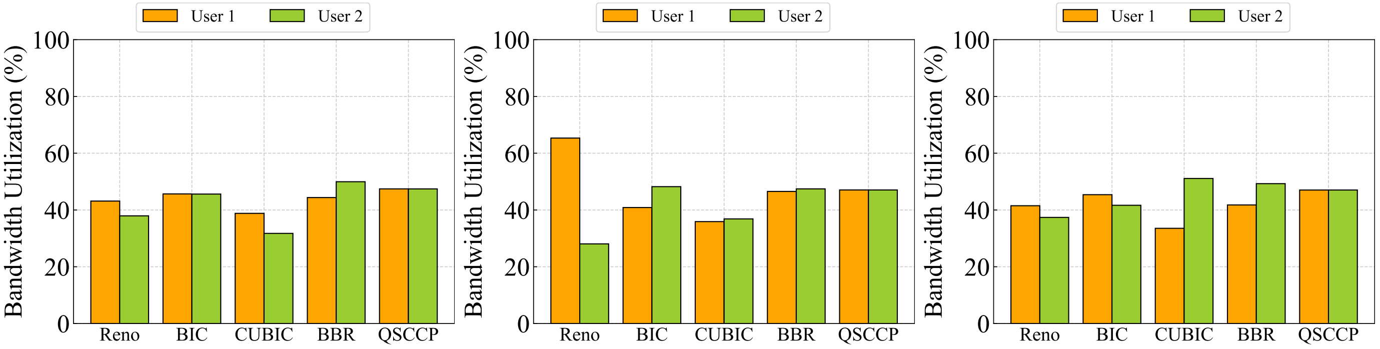
(d) One-way base delay of 60ms (e) One-way base delay of 80ms (f) One-way base delay of 100ms

Fig. 28. Average bandwidth utilization of multiple users during competition with CUBIC under varying latency.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

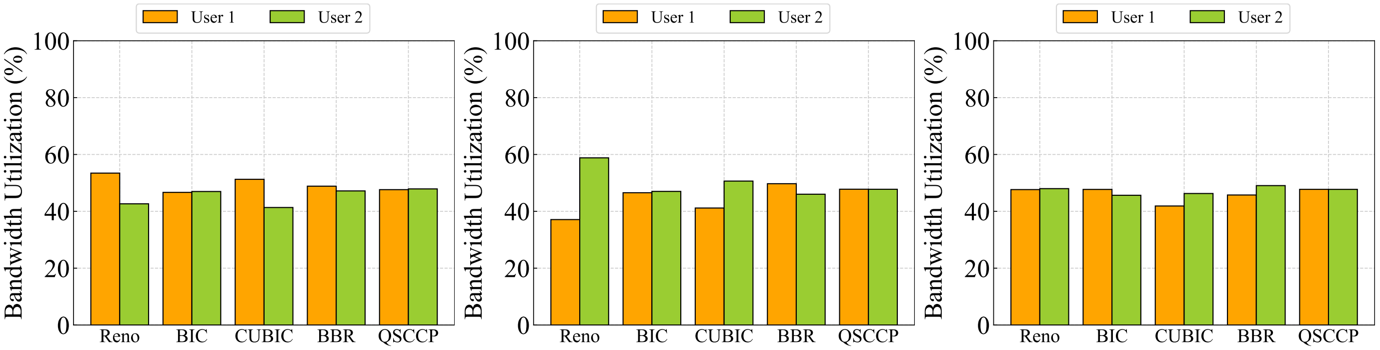


(a) Bottleneck bandwidth of 5Mbps (b) Bottleneck bandwidth of 10Mbps (c) Bottleneck bandwidth of 15Mbps

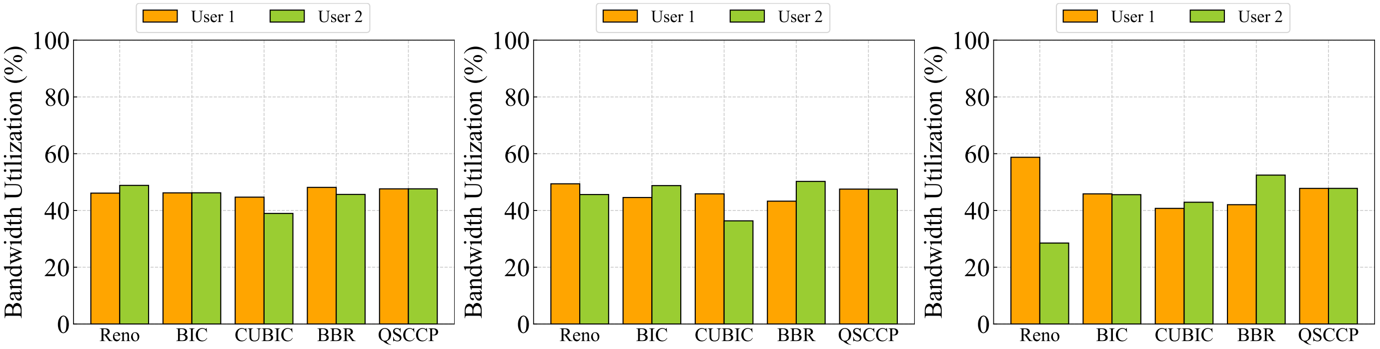


(d) Bottleneck bandwidth of 20Mbps (e) Bottleneck bandwidth of 25Mbps (f) Bottleneck bandwidth of 30Mbps

Fig. 29. Average bandwidth utilization of multiple users using the same algorithm under varying bottleneck bandwidth.



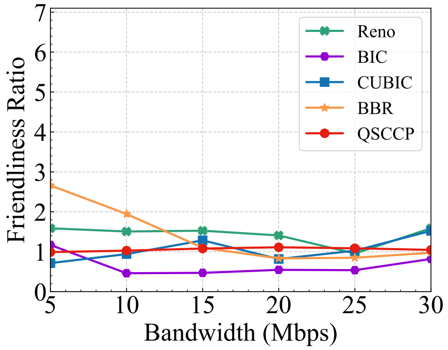
(a) One-way base delay of 10ms (b) One-way base delay of 20ms (c) One-way base delay of 40ms



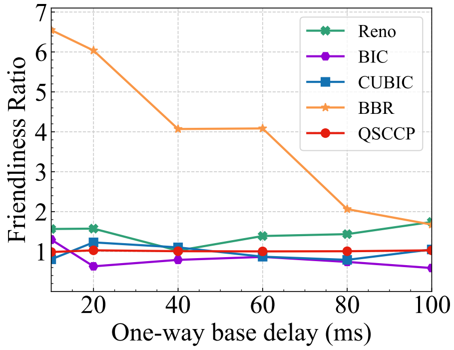
(d) One-way base delay of 60ms (e) One-way base delay of 80ms (f) One-way base delay of 100ms

Fig. 30. Average bandwidth utilization of multiple users using the same algorithm under varying latency.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

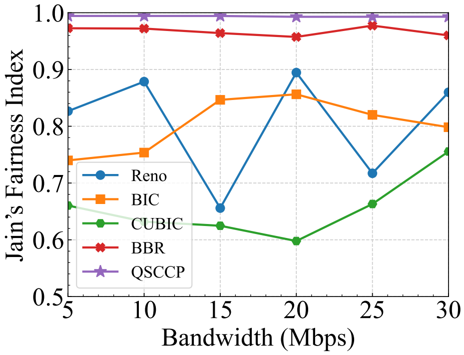


(a) Varying Bandwidth

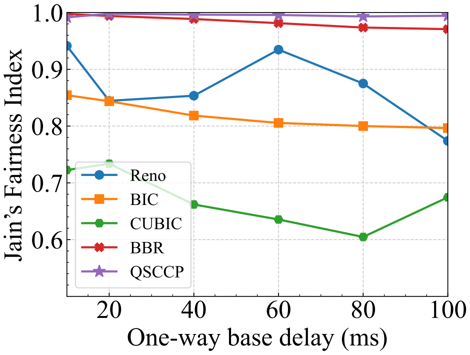


(b) Varying Latency

Fig. 31. Friendliness ratios of five schemes across different bandwidth and latency conditions.



(a) Varying Bandwidth



(b) Varying Latency

Fig. 32. Jain's Fairness Index for each scheme across different bandwidth and latency conditions.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009, pp. 1–12.

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

[3] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[4] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing information networking further: From psirp to pursuit," in *Broadband Communications, Networks, and Systems: 7th International ICST Conference*. Springer, 2012, pp. 1–13.

[5] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf)–an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721–735, 2013.

[6] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, and L. Zhang, "A brief introduction to named data networking," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 1–6.

[7] K. Schneider, C. Yi, B. Zhang, and L. Zhang, "A practical congestion control scheme for named data networking," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 2016, pp. 21–30.

[8] Z. Li, X. Shen, H. Xun, Y. Miao, W. Zhang, P. Luo, and K. Liu, "Coopcon: Cooperative hybrid congestion control scheme for named data networking," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4734–4750, 2023.

[9] S. N. S. Hashemi and A. Bohlooli, "3cp: Coordinated congestion control protocol for named-data networking," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3918–3932, 2021.

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

[11] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, "Congestion control in named data networking–a survey," *Computer Communications*, vol. 86, pp. 1–11, 2016.

[12] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papali, "Multipath congestion control in content-centric networks," in *2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 363–368.

[13] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang, "Optimal multipath congestion control and request forwarding in information-centric networks," in *2013 21st IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2013, pp. 1–10.

[14] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "Mircc: Multipath-aware icn rate-based congestion control," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 2016, pp. 1–10.

[15] Y. Ye, B. Lee, R. Flynn, N. Murray, G. Fang, J. Cao, and Y. Qiao, "Ptp: Path-specified transport protocol for concurrent multipath transmission in named data networks," *Computer Networks*, vol. 144, pp. 280–296, 2018.

[16] 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 Transactions on Networking*, vol. 29, no. 5, pp. 2184–2197, 2021.

[17] F. Wu, W. Yang, M. Sun, J. Ren, and F. Lyu, "Multi-path selection and congestion control for ndn: An online learning approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1977–1989, 2020.

[18] D. Nguyen, M. Fukushima, K. Sugiyama, and A. Tagami, "Efficient multipath forwarding and congestion control without route-labeling in ccn," in *2015 IEEE international conference on communication workshop (ICCW)*. IEEE, 2015, pp. 1533–1538.

[19] J. Yang, Y. Chen, K. Xue, J. Han, J. Li, D. S. Wei, Q. Sun, and J. Lu, "Ieacc: an intelligent edge-aided congestion control scheme for named data networking with deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4932–4947, 2022.

[20] L. Zhang, Y. Cui, M. Wang, K. Zhu, Y. Zhu, and Y. Jiang, "Deepcc: Bridging the gap between congestion control and applications via multiobjective optimization," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2274–2288, 2022.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

[21] "QSCCP Repository," https://github.com/qsccp, 2024 (accessed 24 January 2024).

[22] L. Saino, C. Cocora, and G. Pavlou, "Cctcp: A scalable receiver-driven congestion control protocol for content centric networking," in *2013 IEEE international conference on communications (ICC)*. IEEE, 2013, pp. 3775–3780.

[23] J. Song, M. Lee, and T. T. Kwon, "Smic: Subflow-level multi-path interest control for information centric networking," in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, 2018, pp. 77–87.

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

[25] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for ccn communications," in *2012 Proceedings IEEE INFOCOM Workshops*. IEEE, 2012, pp. 322–327.

[26] M. Nikzad, K. Jamshidi, and A. Bohlooli, "A responsibility-based transport control for named data networking," *Future Generation Computer Systems*, vol. 106, pp. 518–533, 2020.

[27] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An improved hop-by-hop interest shaper for congestion control in named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 55–60, 2013.

[28] S. Mejri, H. Touati, and F. Kamoun, "Hop-by-hop interest rate notification and adjustment in named data networks," in *2018 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2018, pp. 1–6.

[29] H. Touati, S. Mejri, N. Malouch, and F. Kamoun, "Fair hop-by-hop interest rate control to mitigate congestion in named data networks," *Cluster Computing*, vol. 24, pp. 2213–2230, 2021.

[30] W. Yang, Y. Qin, and Y. Yang, "An interest shaping mechanism in ndn: joint congestion control and traffic management," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[31] 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 Computing and Applications*, vol. 31, no. 9, pp. 4675–4683, 2019.

[32] R. T. Braden, D. D. D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, Jun. 1994. [Online]. Available: https://www.rfc-editor.org/info/rfc1633

[33] D. L. Black, Z. Wang, M. A. Carlson, W. Weiss, E. B. Davies, and S. L. Blake, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998. [Online]. Available: https://www.rfc-editor.org/info/rfc2475

[34] D. R. Oran, "Considerations in the Development of a QoS Architecture for CCNx-Like Information-Centric Networking Protocols," RFC 9064, Jun. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9064

[35] X. Ma, A. Afanasyev, and L. Zhang, "A type-theoretic model on ndn-tlv encoding," in *Proceedings of the 9th ACM Conference on Information-Centric Networking*, 2022, pp. 91–102.

[36] Y. Kim, Y. Kim, J. Bi, and I. Yeom, "Differentiated forwarding and caching in named-data networking," *Journal of Network and Computer Applications*, vol. 60, pp. 155–169, 2016.

[37] X. Wang, J. Lv, M. Huang, K. Li, J. Li, and K. Ren, "Energy-efficient icn routing mechanism with qos support," *Computer Networks*, vol. 131, pp. 38–51, 2018.

[38] C. Gündoğan, J. Pfender, P. Kietzmann, T. C. Schmidt, and M. Wählisch, "On the impact of qos management in an information-centric internet of things," *Computer Communications*, vol. 154, pp. 160–172, 2020.

[39] H. Bai, H. Li, J. Que, M. Zhang, and P. H. J. Chong, "Dsccp: A differentiated service-based congestion control protocol for information-centric networking," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 1641–1646.

[40] J. Ros-Giralt, A. Bohara, S. Yellamraju, M. H. Langston, R. Lethin, Y. Jiang, L. Tassiulas, J. Li, Y. Tan, and M. Veeraraghavan, "On the bottleneck structure of congestion-controlled networks," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 3, pp. 1–31, 2019.

[41] J. Ros-Giralt, N. Amsel, S. Yellamraju, J. Ezick, R. Lethin, Y. Jiang, A. Feng, L. Tassiulas, Z. Wu, M. Y. Teh *et al.*, "Designing data center networks using bottleneck structures," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 319–348.

[42] D. R. Oran, "Maintaining CCNx or NDN flow balance with highly variable data object sizes," Internet Engineering Task Force, Internet-Draft draft-oran-icnrg-flowbalance-12, Oct. 2024, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-oran-icnrg-flowbalance/12/

[43] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim 2: An updated ndn simulator for ns-3," *NDN, Technical Report NDN-0028, Revision 2*, 2016. [Online]. Available: https://named-data.net/publications/techreports/ndn-0028-2-ndnsim-v2/

[44] Geant.org, "European topology map (January 2023)," https://resources.geant.org/wp-content/uploads/2023/01/GN4-3N-Topology-Map-Jan2023.pdf, 2023 (accessed 6 September 2023).

[45] S. Charpinel, C. A. S. Santos, A. B. Vieira, R. Villaca, and M. Martinello, "Sdccn: A novel software defined content-centric networking approach," in *2016 IEEE 30Th international conference on advanced information networking and applications (AINA)*. IEEE, 2016, pp. 87–94.

[46] "Mininet," https://github.com/mininet/mininet, 2024 (accessed 6 January 2024).

[47] "Mini-NDN: A Mininet-based NDN emulator," https://github.com/named-data/mini-ndn, 2024 (accessed 6 January 2024).

He Bai received the B.Eng. degree from the School of Information Engineering, Zhengzhou University, China, in 2019. She is currently pursuing the Ph.D. degree from the School of Electronic and Computer Engineering, Peking University, China. Her research interests focus on congestion control and transport protocol.

Hui Li (Senior Member, IEEE) received the B.Eng. and M.S. degrees from Tsinghua University in 1986 and 1989, respectively, and Ph.D. degree from The Chinese University of Hong Kong in 2000. He is a Professor at Peking University, Fellow of IET. He is also the Director of Shenzhen Key Lab of Information theory & Future Internet architecture, the PKU Lab of China Environment for Network Innovations (CENI), National Major Research Infrastructure, Shenzhen Eng. Lab of Converged Networks. His research interests include future network architecture, network protocols, and cyberspace security.

Jianming Que received the B.Eng. degree from the School of Software Technology, Dalian University of Technology, China, in 2019. He received the M.S. degree from the School of Electronic and Computer Engineering, Peking University, China, in 2022. His research interests include network architecture and transport protocol.

Abla Smahi received the B.Sc. and M.Sc degree in Computer Science from Tahri Mohammed University Bechar-Algeria in 2013 and 2015 respectively. She received the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China. She is currently a postdoctoral researcher at Peking University. Her current research includes transport protocol and network security.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2024.3486052

Minglong Zhang (Member, IEEE) received the M.S. degree from Peking University in 2011 and the Ph.D. degree from the Auckland University of Technology, New Zealand, in 2020. He is currently an assistant research professor at Mississippi State University, USA. His research interests include 5G/6G communications, AI-enabled wireless communication and networking, O-RAN networks, and V2X communications.

Peter Han Joo Chong (Senior Member, IEEE) received his Ph.D. degree from the University of British Columbia, Canada, in 2000. He was previously an associate professor (tenured) at Nanyang Technological University, Singapore. He is currently a professor and an associate head of school (research) in the School of Engineering, Computer and Mathematical Sciences at Auckland University of Technology, New Zealand. His research interests include MANETs/VANETs, V2X, artificial intelligence for wireless networks, and 5G networks.

Shuo-Yen Robert Li (Life Fellow, IEEE) received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 1974. He taught applied mathematics with MIT, from 1974 to 1976 and mathematics/statistics/computer science with the University of Illinois, Chicago, IL, USA, from 1976 to 1979. He was with Bell Labs/Bellcore, for a decade and then he joined The Chinese University of Hong Kong, as a Chair Professor, in 1989. He currently serves as the Co-Director of the Institute of Network Coding, and also an Honorary Professor with a few universities. He initiated algebraic switching theory and is a cofounder of network coding theory.

Xiyu Wang received his B.E and M.E. degrees in electrification and automation of railway traction from Beijing Jiaotong University, China, in 1998. He is the Executive Vice President of ZTE Corporation, Shenzhen, China. He is the academic committee director of the National Key Laboratory for Mobile Networks and Mobile Multimedia. His research interests include wireless network technologies.

Ping Lu received his master degree from Southeast University, Nanjing, China, in 1995. He is the Vice President of ZTE Corporation. He is the member of the Big Data Expert Committee of China Computer Society. His research interests include cloud computing and intelligent communications.