



Article

A Single-Rate Multicast Congestion Control (SRMCC) Mechanism in Information-Centric Networking

Yingjie Duan ^{1,2} , Hong Ni ^{1,2}, Xiaoyong Zhu ^{1,2,*} and Xu Wang ^{1,2}

¹ National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences No. 21, North Fourth Ring Road, Haidian District, Beijing 100190, China; duanyingjie17@mails.ucas.ac.cn (Y.D.); nih@dsp.ac.cn (H.N.); wangx@dsp.ac.cn (X.W.)

² School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, China

* Correspondence: zhuxy@dsp.ac.cn; Tel.: +86-1312-116-8320

Abstract: Information-centric networking (ICN) is expected to be a candidate for future internet architecture, and it supports features such as multicast that improves bandwidth utilization and transmission efficiency. However, multicast itself does not provide congestion control. When multiple multicast groups coexist, multicast traffic may exhaust all network resources, and cause network congestion and packet loss. Additionally, traditional IP multicast congestion control mechanisms cannot be directly applied to ICN architecture. Therefore, it is necessary to consider an effective congestion control mechanism for ICN multicast. This paper proposes a single-rate multicast congestion control mechanism, called SRMCC. It supports router-assisted awareness of the network congestion state and congestion control message aggregation. Moreover, the fair shared rate estimation method is innovatively proposed to achieve protocol fairness. Most importantly, it adjusts the rate according to different congestion states indicated by the queue occupancy ratio. By introducing a rate selection factor, it can achieve a balance between packet loss rate and throughput. Experimental results show that our proposal outperforms other mechanisms in throughput, packet loss rate, total bandwidth utilization, and overhead, and achieves protocol fairness and better TCP friendliness.

Keywords: multicast congestion control; single-rate; queue occupancy; ICN



Citation: Duan, Y.; Ni, H.; Zhu, X.; Wang, X. A Single-Rate Multicast Congestion Control (SRMCC) Mechanism in Information-Centric Networking. *Future Internet* **2022**, *14*, 38. <https://doi.org/10.3390/fi14020038>

Academic Editor: Eirini Eleni Tsiropoulou

Received: 20 December 2021

Accepted: 22 January 2022

Published: 25 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Current network architecture and data delivery methods can no longer meet the demand for the efficient transmission of massive data in the future. Researchers are working hard on various supporting technologies, especially the optimization of network architecture [1]. A new network paradigm called information-centric networking (ICN) [2–4] has attracted wide attention.

Unlike host-to-host communication in traditional IP networks, ICN is concerned with the content object itself rather than the location. Separating identity and location, ICN names information and uses the corresponding name rather than the IP address as identification for network traffic. Some ICN solutions have emerged. We are concerned with ICN solutions that can coexist with existing IP networks to achieve incremental deployment and smooth evolution, such as data-oriented network architecture (DONA) [5], MobilityFirst [6], information network (NetInf) [7], and (SEANet) [8]. In such solutions, a name resolution system (NRS) is used to map names to locators and requests are routed to the source based on the locator.

To solve the problems faced in implementing these architectures, many related key techniques are still being investigated, such as naming, routing, caching, multicast [9,10], security, and transport. This paper focuses on the study of multicast transport control techniques, more specifically the multicast congestion control mechanism.

Multicast provides an efficient data transmission method, saving network resources through one-to-many and many-to-many transmission [11]. However, multicast itself does not consider how to deal with changes in network conditions and adjust the transmission rate. When a large amount of multicast traffic exists, the multicast source still transmits data at a fixed rate, which is bound to cause more serious congestion and packet loss. In addition, the multicast delivery stream may be widely distributed across the entire network along the multipoint delivery tree, and if the multicast application cannot respond correctly to network congestion, it will have a more serious impact on the network than the congestion caused by the unicast delivery application [12].

Moreover, the different branch bandwidths of the multicast delivery path and the heterogeneity of receivers make multicast congestion control very difficult [13]. The traditional multicast congestion control mechanism determines the sending rate according to the processing capacity of the worst-performing receiver, which is inefficient and unfair to other receivers. In ICN, some new multicast construction methods have appeared. But as far as we know, there is currently no effective congestion control mechanism that can be applied to ICN multicast. Therefore, it is necessary to study a congestion control mechanism for ICN multicast which can adapt to changes in the network environment and avoid major congestion.

In this paper, we focus on designing an efficient single-rate multicast congestion control mechanism to improve multicast transmission performance. The main contributions of this paper are as follows:

- We design a multicast congestion control mechanism called SRMCC in ICN. SRMCC provides router-assisted awareness of the network congestion state, a congestion control message (MCC) aggregation mechanism, and a single-rate adaptation method. To achieve protocol fairness, a fair shared rate estimation method is also innovatively proposed.
- The single-rate adaptation method proposed in SRMCC adaptively adjusts the multicast rate according to the congestion state indicated by the queue occupancy ratio and introduces a rate selection factor to achieve a balance between packet loss rate and throughput. The definition of network congestion states is provided.
- We develop and implement the SRMCC mechanism in NS-2 [14], and compare it with TFMCC, ASMP, and PerIwithECN. The experimental results prove that SRMCC achieves protocol fairness, improves throughput and total bandwidth utilization, reduces the packet loss rate, and achieves better TCP friendliness than the other three mechanisms. We also verify the effectiveness of the proposed MCC aggregation mechanism by comparing overhead.

The remainder of this paper is organized as follows. We review the related work about multicast congestion control mechanisms in Section 2. Section 3 describes the overall design of SRMCC and its main network elements designed for multicast congestion control. In Section 4, the adaptive rate calculation and adjustment method are described in detail. We then evaluate the performance and discuss the simulation experiments results in Section 5. Finally, we conclude the work in Section 6.

2. Related Work

There are many researchers dedicated to the study of multicast congestion control [15–18]. Most of the mechanisms are based on rate rather than window, because many window-based AIMD algorithms reduce the size of the window by half when they encounter congestion, so that the change in throughput is quite obvious [19]. In contrast, the adjustment of sending rates makes the fluctuation of throughput smoother. Existing rate-based multicast congestion control mechanisms can be divided into two main categories: single-rate congestion control, and multi-rate congestion control [20].

In multi-rate mechanisms, the multicast source maintains multiple layers with different transmission rates, and receivers subscribe to different subsets of layers according to network bandwidth and congestion status [18]. However, these mechanisms are relatively complex and require frequent feedback [15]. Single-rate multicast congestion control

approaches are very effective and easily applied [21], and the mechanism proposed in this paper also belongs to this category. We have focused on various mechanisms which employ the single-rate method and a multicast source to control congestion. For the sake of brevity and novelty, the literature reviews of typical mechanisms in this category are as follows.

Ref. [22] introduced a new single-rate multicast congestion control scheme called explicit rate multicast congestion control (ERMCC). However, ERMCC achieves TCP friendliness only on the path to the slowest receiver with the proper adjustment of its rate adaptation parameters.

Pragmatic general multicast congestion control (PGMCC) [23] is a window-based TCP-like scheme which is based on positive ACKs between the sender and the group representative (the acker). Only the acker is responsible for sending positive ACKs to the multicast source, miming a classic TCP receiver. When packet loss events occur, other receivers send NACKs. In reality, PGMCC does not provide a feedback suppression mechanism.

TCP-friendly multicast congestion control (TFMCC) [24] extends the basic mechanisms of TFRC [25] to dynamically adjust the sending rate. To ensure TCP fairness, TFMCC adopts a conservative method to control the sending rate, which estimates the steady-state throughput of the TCP source according to an equation, and simultaneously uses the round trip time (RTT) and packet loss rate of the slowest receiver mentioned as the current limiting receiver (CLR). However, TFMCC responds slowly to varying networks that cause CLR changes.

Adaptive smooth multicast protocol (ASMP) [26] is a single-rate multicast transport control mechanism designed for multimedia applications. According to the TCP-friendly rate control (TFRC) specification [25], multicast receivers need to calculate the TCP-friendly bandwidth share. The smoothness is achieved by dynamically estimating protocol parameters and adjusting the “smoothness factor”. However, ASMP does not consider the buffer status of the router.

Generalized multicast congestion control (GMCC) [27] uses a set of independent single-rate sub-sessions (also called layers) as building blocks to provide multi-rate features with low complexity. The receiver is responsible for joining and leaving layers. Compared to other multi-rate schemes, GMCC can vary the number of layers and the transmission rate within the layers.

A multicast congestion avoidance scheme named MCA was proposed by [28], which was based on rate and end-to-end. It provides two different schemes based on bit-based and explicit rate-based feedback models. They can avoid the drop-to-zero problem and are fair to unicast congestion avoidance schemes.

Additionally, in the field of ICN, Ref. [29] proposed a lightweight enhancement to the content-oriented publish/subscribe system (R-COPSS) not only for reliability but also for flow and congestion control. It selects an ACKer according to the minimum rate that the publisher can support for the particular pub/sub tree, the packet loss rate, and throughput reported by the subscribers. Then the publisher adjusts the sending rate in a TCP-like way based on the received ACKs. Moreover, a retransmission-based reliable multicast transport for the publish/subscribe internet (PSI) architecture (RMTPSI) [30] was proposed to ensure the reliability of multicast transmission. However, it did not provide a congestion control method, nor did it give an appropriate multicast sending rate. For MobilityFirst architecture, [31] proposed a congestion control mechanism that adopts traffic source rate control and explicit congestion notification from routers. But, at the scale of large networks, the overhead of router feedback would be very large. In addition, some NDN-based congestion control mechanisms have been proposed. NDN supports native multicast delivery because data are able to flow back to the multi-interface at the same time as long as they have the same content name [32]. To avoid congestion, a link quality-based congestion control (LQCC) scheme [33] selects the optimal and stable path according to the defined dynamic metrics related to NDN communication and other metrics such as link utilization, cache store, and the mobility impact. A multipath forwarding strategy for content-centric

network was proposed by [34] for the purpose of fully utilizing available bandwidth. The collaborative multiple metric interface ranking scheme (CM2) [35] combines multiple metrics with different objectives and applies a decision-making function to rank each interface, which can deliver the requested content with less delay and less overhead.

In addition, the new architecture of ICN makes it impossible to directly apply the traditional IP multicast congestion control mechanism. The provision of a congestion control mechanism for ICN multicast is therefore an urgent problem.

3. Design Overview

In this section, we introduce the overall design of the SRMCC mechanism. Firstly, we provide a brief description of ICN multicast architecture and basic multicast transmission. Then, in order to implement SRMCC, we designed the functions related to congestion control for the three main network elements in detail. These include the definition of the multicast congestion control (MCC) message, router-assisted awareness of the network congestion state, the MCC transmission and aggregation mechanism, the estimation of the fair shared rate, and rate adjustment. Note that the rate calculation and adjustment method is described in detail in the Section 4.

3.1. Overview of SRMCC

Figure 1 shows an overview of the SRMCC mechanism. SRMCC is mainly applied to the type of ICN that uses the independent name resolution approach. In this type of ICN, the identifier is completely separated from the locator. Each entity (content, equipment, service, etc.) is assigned an Entity-ID (EID) as an identifier (or name) and a Network Address (NA) as a locator. The Name Resolution System (NRS) dynamically binds the two together. The IP address is adopted as the NA to be compatible with existing IP-based networks. By maintaining both name forwarding information and IP routing information, ICN Routers [36] can forward both ICN packets and ordinary IP packets based on EID and IP address separately. Thus, this type of ICN can coexist with existing IP networks to achieve incremental deployment.

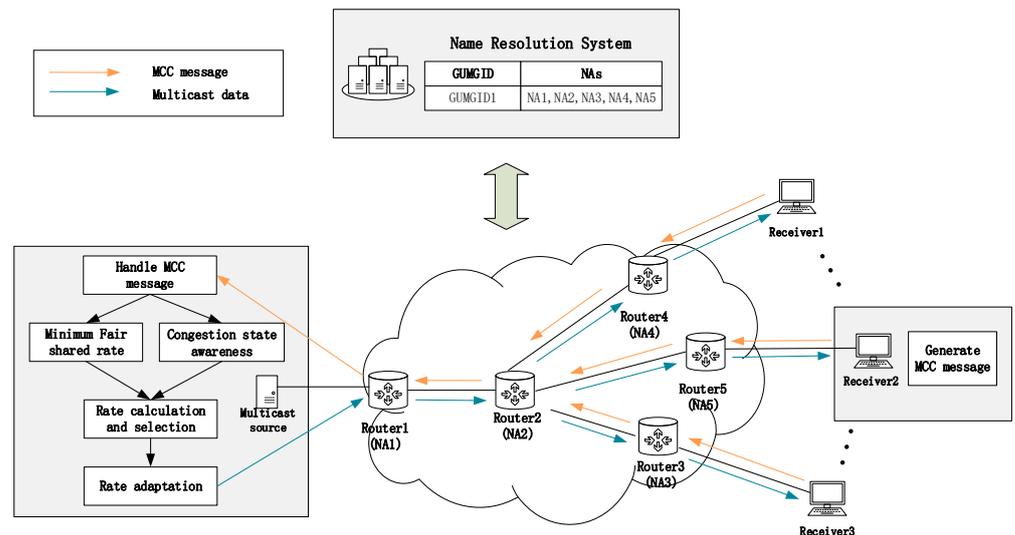


Figure 1. The overview of SRMCC mechanism.

As shown in Figure 1, the multicast system adopted by SRMCC mainly includes NRS, multicast sources, routers, and multicast receivers. The globally unique multicast group identifier (GUMGID) is assigned to every multicast service entity, and NRS maintains the mappings between GUMGIDs and multiple multicast tree node (MTN) NAs, thus enabling the multicast receiver to easily join or leave the multicast service. After the multicast tree is constructed, the multicast data is transmitted along the multicast tree. In the process

of multicast transmission, each MTN keeps a multicast name forwarding table (MNFT) composed of some multicast name forwarding entries (MNFEs). According to the incoming interface of the multicast packet and the GUMGID carried in the multicast packet, the MTN can match the corresponding MNFEs. Then, the multicast data packet is cloned and forwarded out according to the corresponding outgoing interface of matched MNFEs. The focus of this paper is the implementation of congestion control during multicast transmission. The detailed construction process of a multicast tree can be found in [10].

When there are a large number of multicast groups in the network, bottleneck links are inevitable. Since there is no congestion control mechanism, multicast data is still transmitted at a fixed rate. Therefore, congestion and loss may occur, which will affect the performance of multicast transmission.

Based on the ICN multicast system, we designed the SRMCC mechanism to control congestion and loss. In order to perform SRMCC, we designed the feedback message for multicast congestion control, called the MCC message. As shown in Figure 1, the orange line represents MCC messages and the blue line represents multicast data. MCC messages are generated by multicast receivers and transmitted along the reverse path of the multicast tree. The MTN perceives the network congestion state and transmits it to the multicast source through MCC. Moreover, we designed the aggregation mechanism of MCC messages to avoid feedback explosion. After receiving the MCC messages, the multicast source performs rate calculation and selection based on the perceived congestion state and the estimated minimum fair shared rate. The multicast source then adjusts the sending rate, so that congestion can be avoided and each multicast group shares bandwidth fairly.

In the next section, we further introduce the design of SRMCC in detail, including the main network elements and their functions designed for multicast congestion control.

3.2. Main Network Elements

According to Figure 1, this section focuses on the capabilities of the three main network elements (multicast receiver, MTN, and multicast source) designed for multicast congestion control.

The multicast receiver is responsible for periodically generating MCC messages, which are transmitted along the reverse path of the multicast tree. The format of MCC messages is shown in Figure 2. Among them, the initial minimum fair shared rate (*Min fair shared rate*) is set to the link bandwidth C_{access} between the multicast receiver and the access router, and the initial maximum queue occupancy ratio (*Max qocc*) and the initial minimum queue occupancy ratio (*Min qocc*) are set to 0 and 1, respectively. The multicast receiver obtains the C_{access} through the active probe when joining the multicast. This operation only needs to be performed once before multicast transmission and does not need to be performed in real time. Therefore, this operation does not generate overhead in the implementation of SRMCC. Furthermore, the mechanism proposed in this paper is aimed at wired networks.



Figure 2. The format of MCC messages.

The MTN is responsible for perceiving the network state, estimating the fair shared rate, and aggregating MCC messages. Specifically, we designed an MCC table in the MTN to achieve feedback message aggregation. As shown in Figure 3, each MTN maintains an MCC table, which records information on fields relating to the maximum occupancy ratio (*Max qocc*), minimum occupancy ratio (*Min qocc*), and minimum fair shared rate (*Min fair shared rate*) of each multicast group.

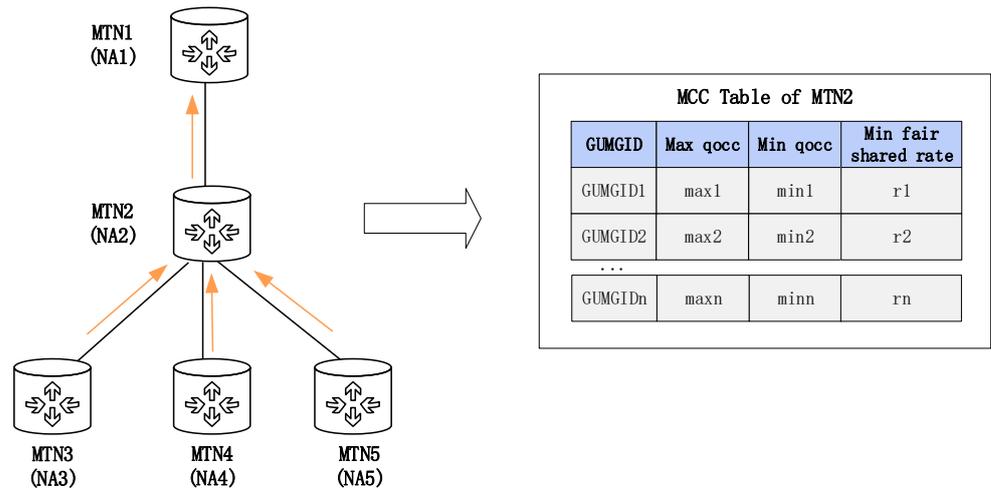


Figure 3. The aggregation of MCC messages.

The multicast tree node MTN_i obtains the current queue length and calculates the queue occupancy ratio $qocc_i$, and then perceives the current network congestion state through a weighted moving average queue occupancy ratio $qocc_{av,i}$,

$$qocc_{av,i} = \alpha \times qocc_i + (1 - \alpha) \times qocc_{av,i-1} \tag{1}$$

where α ($0 < \alpha < 1$) is a weighting factor. $qocc_{av,i}$ is recorded as the maximum occupancy ratio (*Max qocc*) and minimum occupancy ratio (*Min qocc*) in the MCC table of MTN_i .

Inspired by [37,38], we propose a fair shared rate estimation method. The MTN_i checks the number of GUMGID in the MNFT so that it knows the current number of multicast groups. It then randomly samples the received data packets, and different unicast flows are distinguished based on source ID and destination ID, or the source IP and destination IP in IP packets. Thereby, MTN_i gets the estimated total number of flows $N_i(t)$. The ratio of the link bandwidth C_i between the MTN_i and the upstream node to $N_i(t)$ is the estimated fair shared rate, which is recorded as the minimum fair shared rate (*Min fair shared rate*) in the MCC table of MTN_i .

Next, we introduce the MCC messages aggregation mechanism. As shown in Figure 3, when receiving the MCC message from the downstream, the MTN extracts the three fields in the message, and then uses the GUMGID as an index to look up the MCC table. The MTN then compares the values of the three fields in the message with the values recorded in the MCC table. If the *Min fair shared rate* in the message is less than the value recorded in the MCC table, then the *Min fair shared rate* in the MCC table is updated. Similarly, the values of *Max qocc* and *Min qocc* are updated by comparing the *Max qocc* and *Min qocc* in the message and the *Max qocc* and *Min qocc* in the MCC table. A new MCC message is then generated according to the MCC table in the MTN after a period of time, so that only one MCC message is forwarded upstream.

The multicast source initially transmits multicast data at the initial rate, and then dynamically adjusts the sending rate according to the network conditions. According to Equation (2), the multicast source acquires the minimum fair shared rate $R_{fair-shared}$,

$$R_{fair-shared} = \min \left(\frac{C_i}{N_i(t)}, C_{access} \right), i \in N \tag{2}$$

where N is the set of MTNs.

4. Rate Adaptation

In this section, we propose a novel rate adaptation algorithm derived from ALINEA [39], which is a classical traffic control algorithm. We redesign the algorithm and redefine the

meaning of the algorithm parameters in SRMCC. As shown in Equation (3), $R(t+1)_{cal}$ and $R(t)$ mean the calculation rate at time interval $t+1$ and the sending rate at time interval t , respectively:

$$R(t+1)_{cal} = R(t) \times (1 + w \times (O_{th} - O(t))) \quad (3)$$

where, $O(t)$ is the current queue occupancy ratio, w is the weight factor, and O_{th} is the queue occupancy ratio threshold. The values of w and O_{th} are adaptable. At different stages, they are assigned different values to control the fluctuation of variations in the calculation rate. Specifically, the definition of O_{th} is provided in Section 4.1. Moreover, the value of O_{th} and the calculation method of w in different congestion stages are proposed in Section 4.2. Then, the minimum rate $R(t+1)_{min}$ and the maximum rate $R(t+1)_{max}$ for next time interval are determined in Equation (4).

$$R(t+1) = \min(R(t+1)_{cal}, R_{fair-shared}) \quad (4)$$

Next, we define the congestion state value, and then give the rate calculation method corresponding to each congestion state value. Then, we give the rate selection strategy. The overall algorithm pseudo code is shown in Algorithm 1.

4.1. Congestion State Value Definition

In wired links, the queue overflow of routers is the main cause of congestion packet loss. Therefore, we take queue occupancy ratio as the main indicator of current congestion condition. First, based on the queue occupancy condition, we set two queue occupancy ratio thresholds, O_{th1} and O_{th2} . Then, according to the relationship between the current queue occupancy ratio $O(t)$ and the two thresholds, we define four congestion state values (0, 1, 2, 3) corresponding to the stages of no occupancy, light occupancy, moderate occupancy, and heavy occupancy.

$$\text{congestion state value} = \begin{cases} 0, & O(t) = 0 \\ 1, & 0 < O(t) \leq O_{th1} \\ 2, & O_{th1} < O(t) \leq O_{th2} \\ 3, & O(t) > O_{th2} \end{cases} \quad (5)$$

4.2. Rate Calculation

In this section, the rate adjustment method in each stage is introduced in detail. In a round of rate adjustment, the maximum queue occupancy ratio O_{max} and the minimum queue occupancy ratio O_{min} are used, respectively, as the input of the rate calculation module, and the minimum rate $R(t+1)_{min}$ and the maximum rate $R(t+1)_{max}$ are output, respectively. Finally, the final sending rate is obtained according to the rate selection strategy.

4.2.1. No Occupancy Stage

During the startup stage, the multicast source knows nothing about the network. In order to prevent congestion of the network caused by sending a large number of data packets, at this beginning stage, we adopt a conservative method that starts sending at a slow rate. At this stage, the $R(t+1)_{min}$ or $R(t+1)_{max}$ calculation method for each round is as follows:

$$R(t+1) = 2.0 \times R(t) \quad (6)$$

4.2.2. Light Occupancy Stage

After the startup stage, the queue begins to be occupied. Due to the occupancy ratio being relatively low at this time, the sending rate should continue to be increased. As the

occupancy ratio increases, the growth rate of the sending rate should be reduced. Therefore, we introduce a quadratic function to adjust the value of w . At this stage, O_{th} is set as O_{th1} .

$$w = 1 - \frac{3}{4} \times \frac{O(t)^2}{O_{th1}^2} \tag{7}$$

Then $R(t + 1)_{min}$ or $R(t + 1)_{max}$ are adjusted for the next round according to Equations (3) and (4).

4.2.3. Moderate Occupancy Stage

When the queue occupancy ratio is larger than O_{th1} and less than O_{th2} , network congestion is likely to occur, so the growth rate of the sending rate should be reduced faster than during the light occupancy stage. To do this, we introduce a linear function to linearly reduce w . At the same time, in order to improve throughput, the rate is still not reduced at this stage.

$$w = \frac{1}{4} \times \frac{O(t) - O_{th2}}{O_{th1} - O_{th2}} \tag{8}$$

Then $R(t + 1)_{min}$ or $R(t + 1)_{max}$ are adjusted for the next round according to Equations (3) and (4).

4.2.4. Heavy Occupancy Stage

At this time, the queue occupancy ratio has exceeded the maximum threshold O_{th2} , and network congestion is prone to occur or has occurred. In order to alleviate network congestion, the usual approach is to decrease the rate immediately. However, if the current queue is mainly occupied by other competing streams, continuously reducing the multicast sending rate will cause the multicast throughput to drop-to-zero. To maintain fairness, it is necessary to determine whether the heavy occupancy is caused by the multicast traffic itself. We innovatively introduce the minimum fair shared rate $R_{fair-shared}$ as the demarcation, and the calculation rate for the next time interval is adjusted according to Equation (9).

$$R(t + 1)_{cal} = \begin{cases} \min\left(R(t + 1)_{cal}, \frac{1}{2} \times R(t)\right), & R(t) \geq R_{fair-shared} \\ 2.0 \times R(t), & R(t) < R_{fair-shared} \end{cases} \tag{9}$$

As shown in Equation (9), when $R(t)$ is larger than or equal to $R_{fair-shared}$, the heavy occupancy state is caused by multicast traffic. Therefore, the multicast rate should be reduced to alleviate congestion. At this time, $(O_{th} - O(t))$ is a negative value, and w is set to 1.0 in Equation (3) to speed up the decline of the calculation rate. When $R(t)$ is less than $R_{fair-shared}$, the situation is caused by other competing traffic, and the multicast rate should be increased to preempt bandwidth and prevent throughput from dropping to zero.

Finally, $R(t + 1)_{min}$ or $R(t + 1)_{max}$ are adjusted for the next round according to Equation (4).

4.3. Rate Selection Strategy

Unlike other classic IP multicast congestion control methods, we do not adjust the multicast rate according to the receiver with worst receiving capability. By introducing the rate selection factor γ , we make full use of the buffer of router to improve throughput. The optimal packet loss rate and throughput can be achieved simultaneously by setting the value of γ properly.

According to the rate calculation method in the last section, the minimum rate $R(t + 1)_{min}$ and the maximum rate $R(t + 1)_{max}$ are calculated based on the maximum queue occupancy ratio O_{max} and the minimum queue occupancy ratio O_{min} . According to Equation (10), the final sending rate for the next interval can be determined.

$$R(t + 1) = R(t + 1)_{min} + \gamma \times (R(t + 1)_{max} - R(t + 1)_{min}) \tag{10}$$

Algorithm 1. The Adaptation Method of Rate**Input:** $O(t)$, $R_{fair-shared}$ **Output:** $R(t+1)$

```

1: initialization:  $O_{th1} = 0.25$ ,  $O_{th2} = 0.5$ ,  $\gamma = 0.15$ ,  $R(t+1)_{min} = R(t+1)_{max} = 0$ 
2: for  $O(t)$  in  $\{O_{min}, O_{max}\}$  do
3:   if  $O(t) = 0$  then
4:     congestion state value = 0
5:     calculate  $R(t+1)_{min}$  or  $R(t+1)_{max}$  using Equation (6)
6:   else if  $0 < O(t) \leq O_{th1}$  then
7:     congestion state value = 1,  $O_{th} = O_{th1}$ ,  $w = 1 - \frac{3}{4} \times \frac{O(t)^2}{O_{th1}^2}$ 
8:      $R(t+1)_{cal} = R(t) \times (1 + w \times (O_{th} - O(t)))$ 
9:     calculate  $R(t+1)_{min}$  or  $R(t+1)_{max}$  using Equation (4)
10:  else if  $O_{th1} < O(t) \leq O_{th2}$  then
11:    congestion state value = 2,  $O_{th} = O_{th2}$ ,  $w = \frac{1}{4} \times \frac{O(t) - O_{th2}}{O_{th1} - O_{th2}}$ 
12:     $R(t+1)_{cal} = R(t) \times (1 + w \times (O_{th} - O(t)))$ 
13:    calculate  $R(t+1)_{min}$  or  $R(t+1)_{max}$  using Equation (4)
14:  else if  $O(t) > O_{th2}$  then
15:    congestion state value = 3,  $O_{th} = O_{th2}$ ,  $w = 1$ 
16:    if  $R(t) \geq R_{fair-shared}$  then
17:       $R(t+1)_{cal} = R(t) \times (1 + w \times (O_{th} - O(t)))$ 
18:       $R(t+1)_{cal} = \min(R(t+1)_{cal}, \frac{1}{2} \times R(t))$ 
19:      calculate  $R(t+1)_{min}$  or  $R(t+1)_{max}$  using Equation (4)
20:    else if  $R(t) < R_{fair-shared}$  then
21:       $R(t+1)_{cal} = 2.0 \times R(t)$ 
22:      calculate  $R(t+1)_{min}$  or  $R(t+1)_{max}$  using Equation (4)
23:    end if
24:  end if
25: end for
26:  $R(t+1) = R(t+1)_{min} + \gamma \times (R(t+1)_{max} - R(t+1)_{min})$ 
27: return  $R(t+1)$ 

```

5. Evaluation

This section describes the series of experiments that we conducted to evaluate the performance of the proposed mechanism. We developed SRMCC and PerIfwithECN [31] over NS-2 [14] and used the implementations for TFMCC [24] and ASMP [26] that were previously developed in NS-2. First, we performed basic technical verification on SRMCC, then compared SRMCC with TFMCC, ASMP, and PerIfwithECN with regard to throughput, bandwidth utilization, and packet loss rate. Finally, we compared the overhead of the four mechanisms.

5.1. Simulation Setup

The simulation topology is shown in Figure 4. We set various experiment scenarios to investigate the performance. In scenarios 1 and 2, the bandwidths of the links between R0–R1 were set to 5 Mbps and 2 Mbps, respectively. And in scenario 3, the bandwidths of the links between R0–R1 and between R1 and each receiver were set to 2 Mbps and 1.5 Mbps, respectively. In the above three scenarios, there was one multicast traffic and two TCP traffic. Then, in scenario 4, we tested the responsiveness of SRMCC, TFMCC, ASMP, and PerIfwithECN to changes of competing traffic, and the bandwidth of the link between R0–R1 was set to 2 Mbps. According to [26], the bottleneck link was shared by one multicast stream, two TCP flows, and one constant bit rate (CBR) applications. At the beginning of the simulation, the first TCP flow (TCP1) was transmitted with the multicast stream. At the 30th second, the second TCP flow (TCP2) started transmission. And it stopped transmission at the 120th second. The CBR application started transmission at 600 kb/s at the 80th second, and stopped transmission at the 170th second.

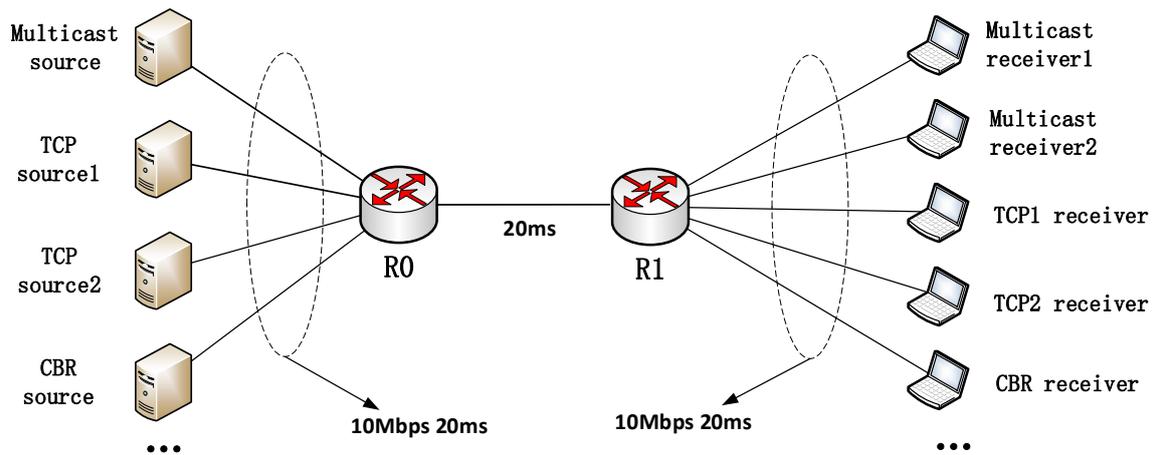


Figure 4. The simulation topology.

Note that since we considered a hybrid network of IP and ICN and the entire network consisted of ICN routers and IP routers, TCP was the traffic in the network that could not be ignored. To compare performance with other mechanisms and evaluate the friendliness of these mechanisms to TCP, we introduced TCP as the primary competitive traffic. Additionally, Table 1 shows the experiment parameters and their values for our simulation.

Table 1. Experiment parameters.

Parameter	Value
The maximum queue length (Q_{max})	50 packets
The threshold 1 of queue occupancy ratio (O_{th1})	0.25
The threshold 2 of queue occupancy ratio (O_{th2})	0.50
Weighting factor α	0.9
Packet size	1000 bytes
Rate selection factor (γ)	0.15
Rate adaptation cycle	$4.0 \times RTT$
Simulation time	200 s
The rate of CBR application	600 Kb/s
Initial rate of multicast	50 Kb/s

5.2. Basic Performance of SRMCC

5.2.1. Experiments on the Setting of Value γ

We performed a series of experiments to explore the effect of parameter γ on SRMCC performance in the first three scenarios. Figure 5a represents the ratio of the multicast throughput to TCP throughput under different rate selection factors γ , and Figure 5b represents the packet loss rate of multicast traffic under different rate selection factors γ . As shown in Figure 5, as the value of γ increases, both the throughput ratio of multicast traffic using SRMCC to TCP traffic and the packet loss rate are on the rise. Figure 5 shows that SRMCC can achieve good performance in a relatively wide range of γ . In particular, when $\gamma = 0.15$, the throughput ratio is closest to 1 and the packet loss rate is equal to or close to 0, which can simultaneously achieve optimal performances when it comes to TCP friendliness and packet loss rate. Therefore, we set γ to 0.15 in our experiments.

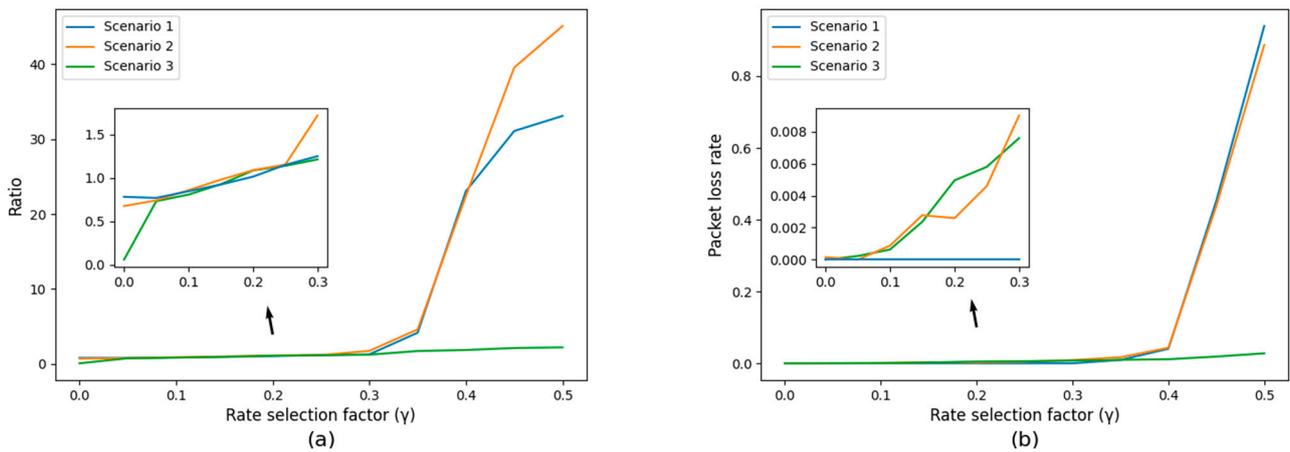


Figure 5. Experiments on the setting of value γ . (a) The throughput ratio of multicast traffic to TCP traffic under different factors rate selection γ ; (b) The packet loss rate of multicast traffic under different rate selection factors γ .

5.2.2. The Basic Performance of SRMCC

The proposed rate adaption method is based on fair shared rate estimation, for which we performed many experiments to evaluate the performance of SRMCC, in terms of throughput, packet loss rate, and bandwidth utilization, in the case of multiple multicast groups co-existing.

In Figure 6a, we take four multicast flows as an example to show the variation in throughput of each multicast flow over time in the presence of multiple multicast groups. We can see that after a short period of adjustment, the throughput of the four multicast flows becomes very even and does not fluctuate very much. That reflects the aggregation of the proposed rate adaptation method and the effectiveness of the fair shared rate estimation method. Figure 6b shows variations in the average packet loss rate of SRMCC under different numbers of multicast groups. As the number of multicast groups increases, the packet loss rate remains at a relatively low level, which can meet the reliability requirements of most applications. Figure 6c shows the total bandwidth utilization of SRMCC under different numbers of multicast groups. The total bandwidth utilization hardly decreases as the number of multicast groups increases, which reflects the good scalability of SRMCC.

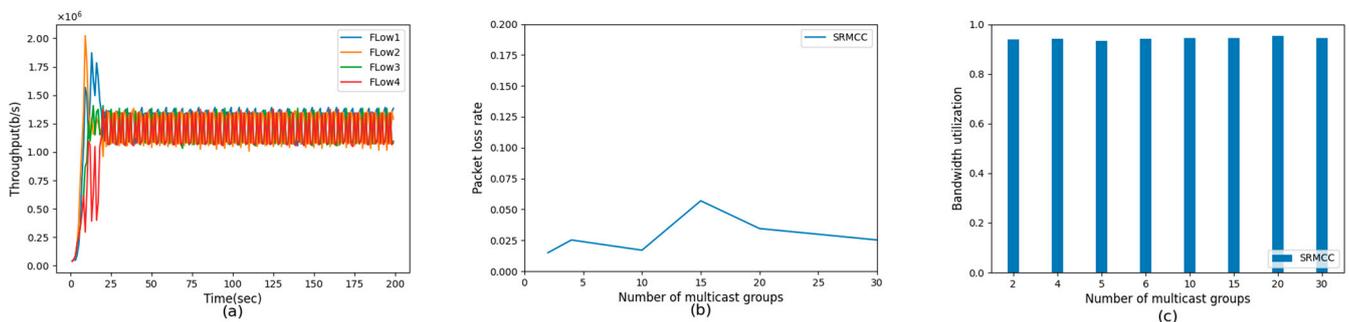


Figure 6. The basic performance of SRMCC. (a) The variation in throughput over time under four multicast groups; (b) The variation in packet loss rate under different numbers of multicast groups; (c) The variation in total bandwidth utilization under different numbers of multicast groups.

In summary, SRMCC is able to achieve good protocol fairness, scalability, and a low packet loss rate, and can be well applied to scenarios where multiple multicast groups exist.

5.3. Performance Comparison with ASMP, TFMCC, and PerIwithECN

5.3.1. Throughput

Figure 7 presents the achieved throughput of multicast receivers of each mechanism under different bottleneck link bandwidths and the scenario of dynamic competing traffic, respectively. Figure 8 presents the comparison of average throughput of the multicast and TCP under different scenarios.

From Figure 7, it can be seen that the throughput of SRMCC is significantly improved, and SRMCC responds faster to the dynamic changes of competing traffic than the other three mechanisms. Compared with ASMP, the throughput of SRMCC is improved by 11.3%, 8.3%, 0.9% in scenarios 1–3, respectively. The throughput of TFMCC is the largest. However, in Figure 8, except for in scenario 4, TFMCC consumes a higher bandwidth share than TCP connections. For example, we observe that TFMCC consumes on average 151.8% of the expected bandwidth share, while a TCP only consumes 72.6% of its share under a bottleneck link bandwidth of 5 Mbps. Our assessment is that, given the multiple TFMCC streams in the internet, TCP will suffer from starvation because TFMCC consumes more bandwidth than TCP streams when sharing the same bottleneck link. It can be seen that TFMCC is the least TCP-friendly, and the increase in its throughput comes at the expense of TCP’s throughput.

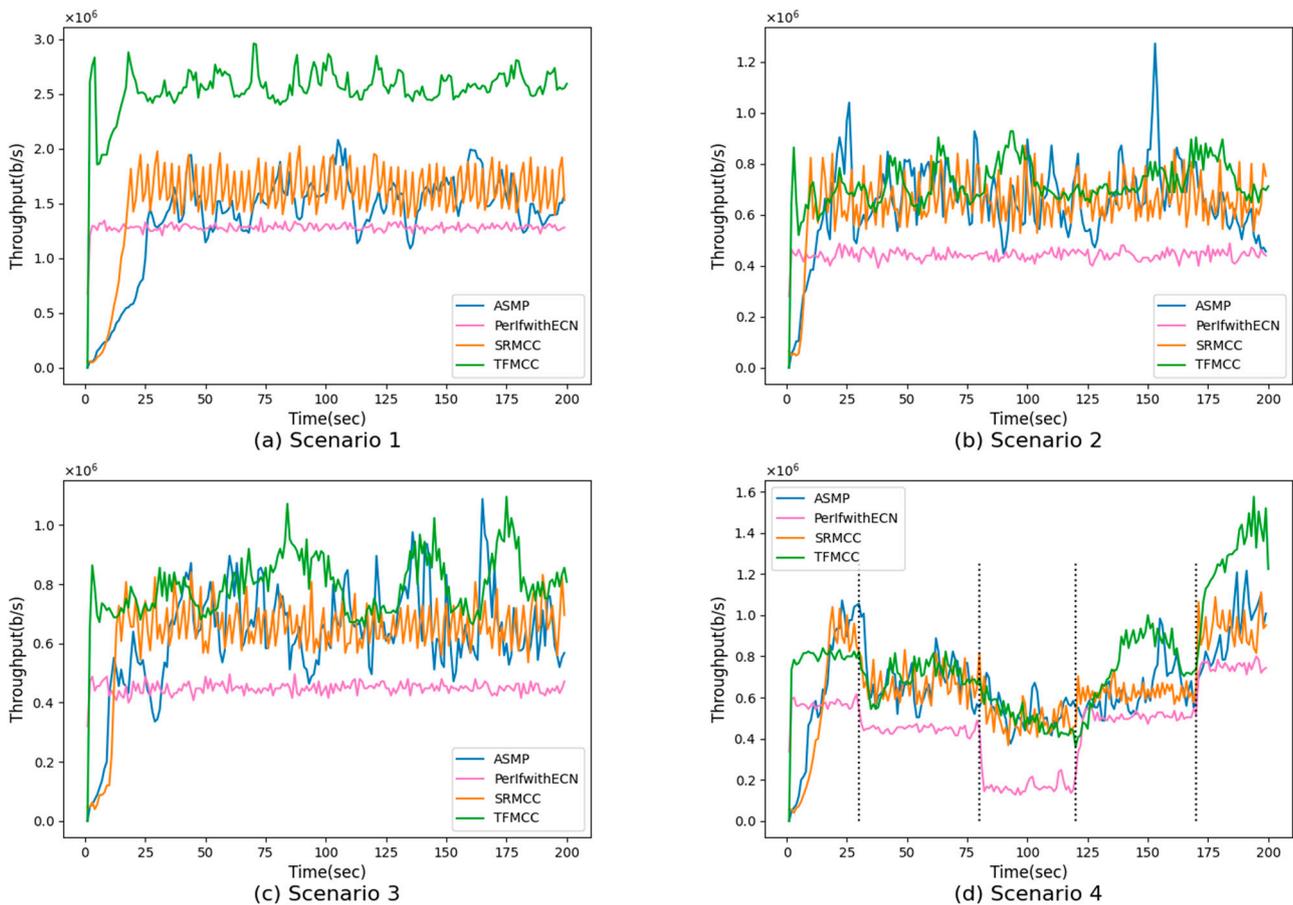


Figure 7. The variation in throughput over time under different scenarios. (a) The variation in throughput over time under scenario 1; (b) The variation in throughput over time under scenario 2; (c) The variation in throughput over time under scenario 3; (d) The variation in throughput over time under scenario 4.

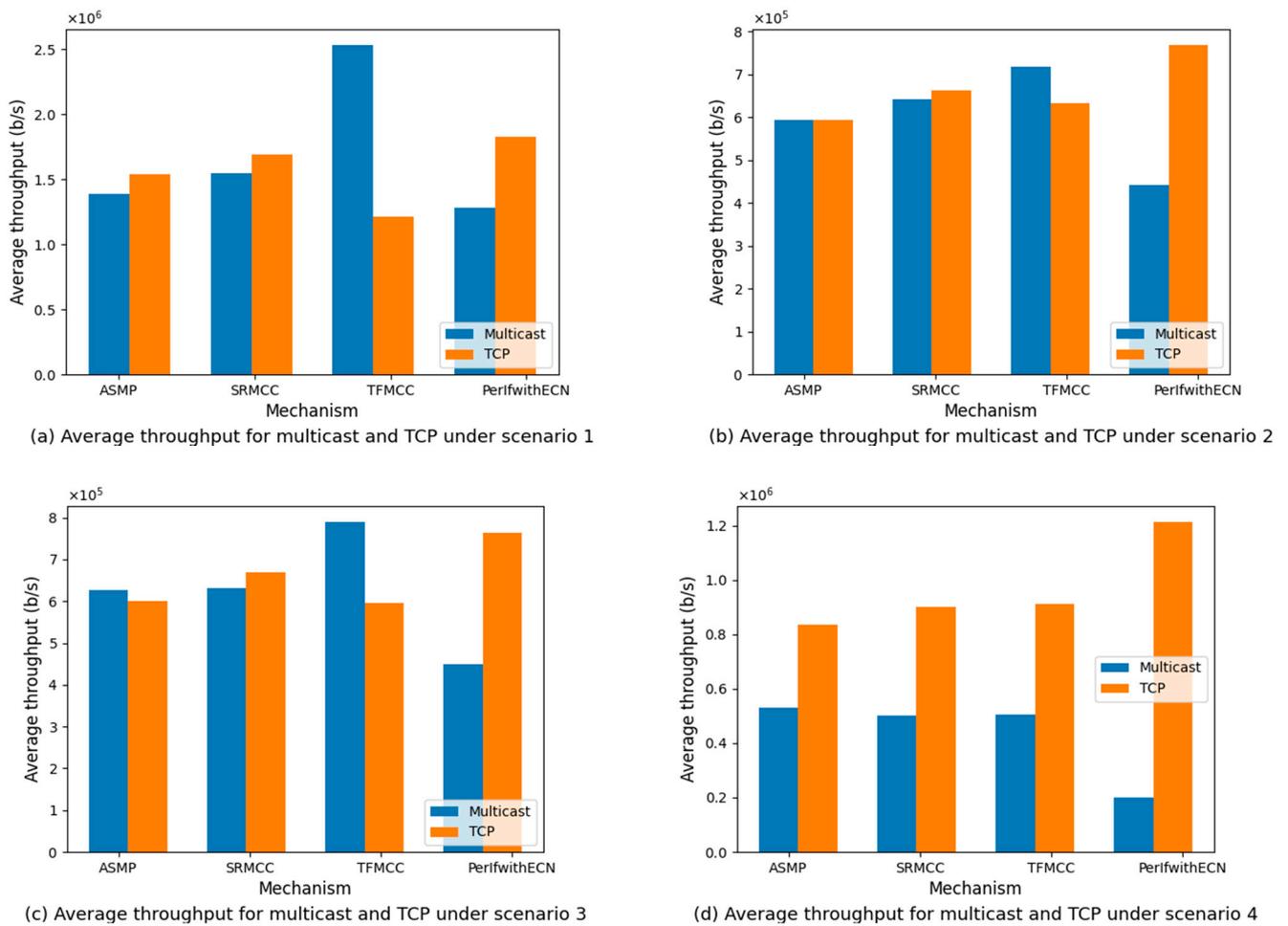


Figure 8. The comparison of average throughput of multicast and TCP under different scenarios. (a) The comparison of average throughput of multicast and TCP under scenario 1; (b) The comparison of average throughput of multicast and TCP under scenario 2; (c) The comparison of average throughput of multicast and TCP under scenario 3; (d) The comparison of average throughput of multicast and TCP under scenario 4.

In contrast, ASMP and SRMCC are more TCP-friendly than TFMCC. ASMP consumes less bandwidth than TCP, except in the scenario 3. In all of the experiments, SRMCC consumes less bandwidth share than TCP connections. Therefore, SRMCC is more TCP-friendly than ASMP. Moreover, it is worth noting that, as can be seen from Figure 8, it is not only the throughput of SRMCC which is improved, but also the throughput of TCP. However, in the four scenarios, the throughput of multicast under PerIfwithECN is significantly lower than that of TCP. Since the router detects congestion, it will display a notification to the source to slow down, and TCP will preempt more bandwidth resources, resulting in extremely low multicast throughput.

Overall, SRMCC achieves higher throughput and improves transmission efficiency while improving TCP friendliness.

5.3.2. Bandwidth Utilization

Bandwidth utilization is an important indicator that should not be ignored. To evaluate transmission performance, we also tested the bandwidth utilization of multicast traffic and total bandwidth utilization of all traffic under different bottleneck link bandwidths and scenarios. Owing to the fact that each competing stream in scenario 4 started and stopped at different times, we compared the bandwidth utilization between the 80th second and the 120th second, since four streams coexisted during this period.

On the one hand, from Figure 9b, it can be seen that the proposed SRMCC mechanism can achieve almost the same total bandwidth utilization as TFMCC and PerIfwithECN under different scenarios, and the total bandwidth utilization of SRMCC is 7.1–9.5% higher than that of ASMP under the same link conditions.

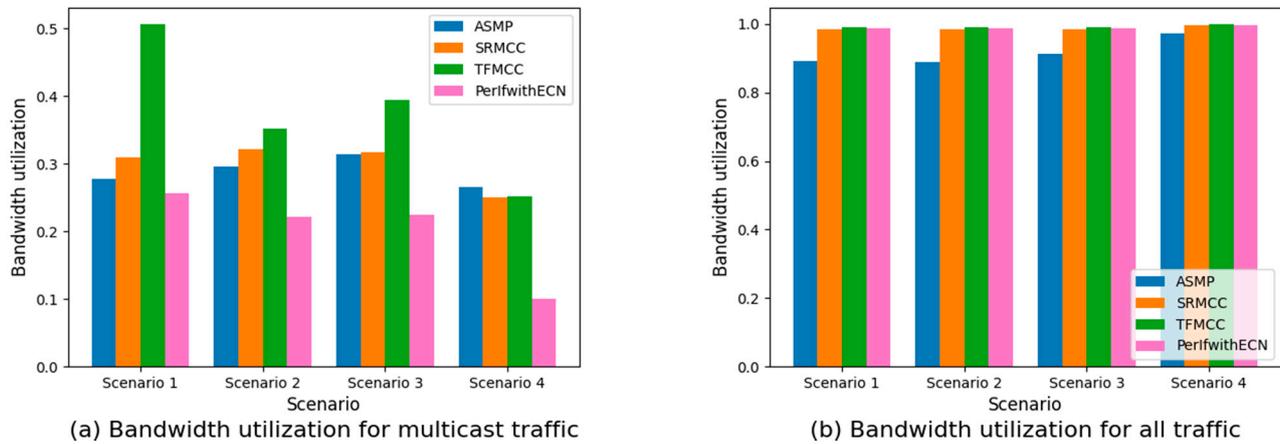


Figure 9. The variation in bandwidth utilization under different scenarios. (a) The bandwidth utilization of multicast traffic under different scenarios; (b) The total bandwidth utilization of all traffic under different scenarios.

On the other hand, in Figure 9a, the bandwidth utilization of TFMCC is 19.7%, 3.8%, and 7.8% higher than that of SRMCC in scenarios 1, 2 and 3, respectively. However, the total bandwidth utilization of the two is almost the same, and the difference is less than 1%. This is because when TFMCC and TCP coexist, TFMCC greatly squeezes out TCP traffic, which also verifies the conclusion of the last section that TFMCC is extremely unfriendly to TCP. However, the situation with PerIfwithECN is the opposite of TFMCC. Although the total bandwidth utilization of PerIfwithECN and SRMCC is similar, the former is mainly contributed by TCP, and the bandwidth utilization of multicast streams under PerIfwithECN is extremely low. Therefore, the multicast under PerIfwithECN cannot coexist with TCP fairly.

Moreover, taking into account the conclusions of the last section, we can conclude that SRMCC achieves almost the same total bandwidth utilization as TFMCC and PerIfwithECN under the same link conditions, while maintaining better TCP friendliness than ASMP. In addition, it can be seen from Figure 8 that the increase in total bandwidth utilization of SRMCC is contributed to by both SRMCC and TCP.

5.3.3. Packet Loss Rate

We compared the packet loss rate for the mechanisms under the four scenarios. As shown in Figure 10, the packet loss rates of SRMCC and PerIfwithECN are almost 0 in all experiment scenarios. However, the packet loss rates of TFMCC and ASMP are higher. In particular, except for in scenario 1, the packet loss rate of ASMP is about 2%. It can be seen that the proposed SRMCC mechanism can significantly reduce the packet loss rate, and this is because it adjusts the rate based on the congestion state indicated by the queue occupancy ratio, while also controlling the queue occupancy ratio by introducing a rate selection factor. Keeping in mind the conclusions of Section 5.3.1, it can be verified that by reasonably setting the value of the rate selection factor, a balance between packet loss rate and throughput can be achieved.

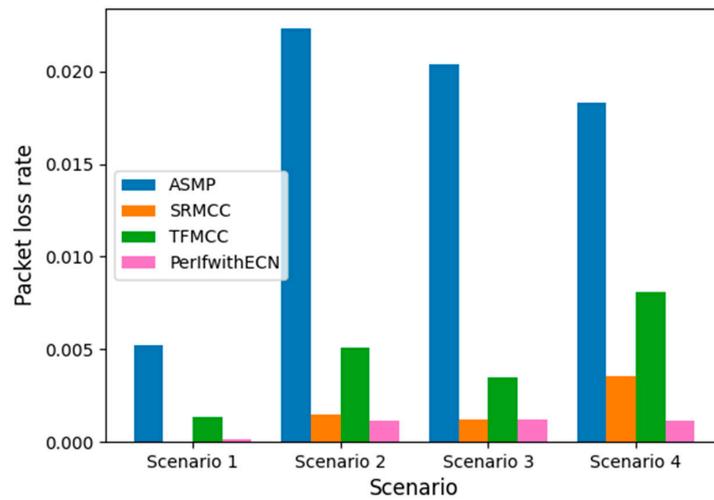


Figure 10. The variation in packet loss rate under different scenarios.

5.3.4. Overhead Analysis

In order to perform multicast congestion control, various mechanisms bring additional overhead. Due to the fact that these mechanisms adjust the sending rate based on the feedback information, we analyzed overhead with this aspect in mind.

Figure 11 shows the number of feedback packets received by the multicast source of each multicast mechanism under different scenarios. It can be seen that SRMCC achieves the smallest number of feedback packets in all scenarios, while PerfIfwithECN achieves the largest number of feedback packets, followed by TFMCC. This is because there is an MCC table in each MTN to perform feedback aggregation to ensure that only one feedback packet is sent upstream. However, in TFMCC, in addition to the current limiting receiver (CLR) sending continuous instant feedback to the source, any receiver whose expected throughput is lower than the current rate of the source also sends feedback packets. For PerfIfwithECN, once the outgoing interface queue of the router detects that the congestion condition is met, the router immediately sends the suggested rate to the corresponding source node. Furthermore, seeing as there is no feedback aggregation mechanism, it is conceivable that with the expansion of the scale of the network, the overhead of PerfIfwithECN would be greater. Moreover, the number of feedback packets of SRMCC decreases by an average of 82.4%, 50.8%, and 94.1% compared with TFMCC, ASMP, and PerfIfwithECN, respectively. It can be proved that the proposed MCC message aggregation mechanism is effective.

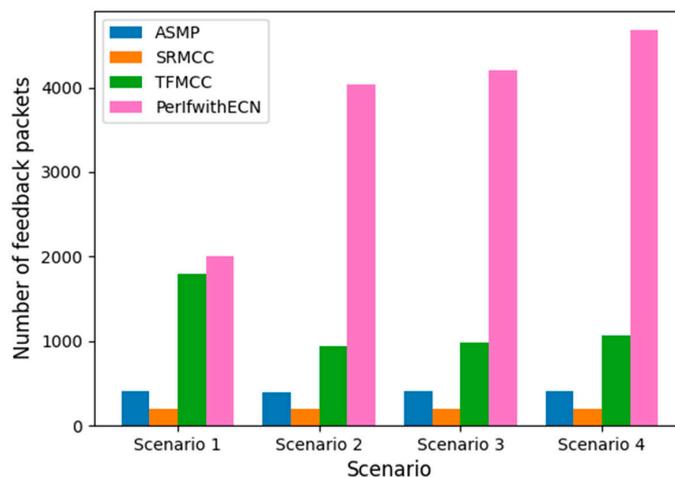


Figure 11. The variation in the number of feedback packets under different scenarios.

In summary, SRMCC not only performs better than other classic multicast congestion control mechanisms in terms of other indicators, but it also reduces overhead. Therefore, the proposed SRMCC is effective.

6. Conclusions

In this paper, a single-rate multicast congestion control mechanism called SRMCC was proposed for ICN. It can adaptively increase or decrease rates when the network is underutilized or about to become congested. To this end, we designed the format of MCC messages, a router-assisted awareness of the network congestion state, MCC message transmission and aggregation mechanism. In order to adjust the rate reasonably, we introduced a rate selection factor. Moreover, we innovatively proposed a fair shared rate estimation method to achieve protocol fairness. The simulation results demonstrate the significant advantages of the proposal in this paper over other mechanisms in throughput, total bandwidth utilization, packet loss rate, and overhead. In addition, our proposal maintains protocol fairness and improves TCP friendliness.

In the future, important issues, such as how to extend this mechanism to scenarios with variable access bandwidth, such as wireless networks, will be explored. In addition, we will further explore the combination of a multicast congestion control mechanism and reliable multicast protocol to provide better transmission performance.

Author Contributions: Conceptualization, Y.D., H.N. and X.Z.; methodology, Y.D., H.N. and X.Z.; software, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D., H.N., X.Z. and X.W.; supervision, X.Z.; project administration, X.Z.; funding acquisition, H.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Strategic Leadership Project of Chinese Academy of Sciences: SEANET Technology Standardization Research System Development (Project No. XDC02070100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: We would like to express our gratitude to Jinlin Wang, Rui Han, and Zhiyuan Wang for their meaningful support of this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xylomenos, G.; Ververidis, C.N.; Siris, V.A.; Fotiou, N.; Tsilopoulos, C.; Vasilakos, X.; Katsaros, K.V.; Polyzos, G.C. A survey of information-centric networking research. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1024–1049. [[CrossRef](#)]
2. Ahlgren, B.; Dannewitz, C.; Imbrenda, C.; Kutscher, D.; Ohlman, B. A survey of information-centric networking. *IEEE Commun. Mag.* **2012**, *50*, 26–36. [[CrossRef](#)]
3. Jiang, X.; Bi, J.; Nan, G.; Li, Z. A survey on Information-centric Networking: Rationales, designs and debates. *China Commun.* **2015**, *12*, 1–12. [[CrossRef](#)]
4. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking Named Content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Italy, Rome, 1–4 December 2009; pp. 1–12.
5. Koponen, T.; Chawla, M.; Chun, B.-G.; Ermolinskiy, A.; Kim, K.H.; Shenker, S.; Stoica, I. A data-oriented (and beyond) network architecture. In Proceedings of the ACM SIGCOMM Computer Communication Review, Kyoto, Japan, 27 August 2007; pp. 181–192.
6. Raychaudhuri, D.; Nagaraja, K.; Venkataramani, A. MobilityFirst: A robust and trustworthy mobility-centric architecture for the future internet. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2012**, *16*, 2–13. [[CrossRef](#)]
7. Dannewitz, C.; Kutscher, D.; Ohlman, B.; Farrell, S.; Ahlgren, B.; Karl, H. Network of Information (NetInf)—An informationcentric networking architecture. *Comput. Commun.* **2013**, *36*, 721–735. [[CrossRef](#)]
8. Wang, J.; Chen, G.; You, J.; Sun, P. SEANet: Architecture and Technologies of an On-site, Elastic, Autonomous Network. *J. Netw. New Media.* **2020**, *9*, 1–8.
9. Yang, B.; Chen, X.; Xie, J.; Li, S.; Yang, J. Multicast Design for the MobilityFirst Future Internet Architecture. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019.

10. Li, B.; Wang, J. An Identifier and Locator Decoupled Multicast Approach (ILDM) Based on ICN. *Appl. Sci.* **2021**, *11*, 578. [CrossRef]
11. Lao, L.; Cui, J.H.; Gerla, M.; Maggiorini, D. A comparative study of multicast protocols: Top, bottom, or in the middle? In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; pp. 2809–2814.
12. Whetten, B.; Conlan, J.; Communications, G. A Rate Based Congestion Control Scheme for Reliable Multicast. *Technical White Paper, GlobalCast Communications*. 1998. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.6783&rep=rep1&type=pdf> (accessed on 18 December 2021).
13. Chakraborty, D.; Chakraborty, G.; Shiratori, N. A dynamic multicast routing satisfying multiple QoS constraints. *Int. J. Netw. Manag.* **2003**, *13*, 321–335. [CrossRef]
14. NS-2 Simulator. Available online: <https://www.isi.edu/nsnam/ns/ns-build.html> (accessed on 4 November 2005).
15. Singhal, N.; Sharma, R.M. Survey on TCP Friendly Congestion Control for Unicast and Multicast Traffic. *Int. J. Comput. Appl.* **2011**, *26*, 23–30. [CrossRef]
16. Matrawy, A.; Lambadaris, I. A survey of congestion control schemes for multicast video applications. *Commun. Surv. Tutor. IEEE* **2003**, *5*, 22–31. [CrossRef]
17. Kammoun, W.; Youssef, H. An Adaptive Mechanism for End-to-End Multirate Multicast Congestion Control. In Proceedings of the 2008 the Third International Conference on Digital Telecommunications (ICDT 2008), Bucharest, Romania, 29 June–5 July 2008; pp. 88–93.
18. Huo, L.; Yi, J. Research on Multicast Congestion Control. In Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 August 2015; pp. 846–850.
19. Zhao, J.; Yang, M.; Zhang, F. Review of multicast Congestion Control. *J. Small Microcomput. Syst.* **2004**, *25*, 511–518.
20. Shi, F.; Wu, J. Summary of Multicast Congestion Control. *J. Softw.* **2002**, *13*, 1441–1448.
21. Kumar, S.; Singh, K. Logarithmic Based Multicast Congestion Control Mechanism. In Proceedings of the Industrial and Intelligent Information (ICIII 2012), Singapore; 2012; pp. 102–108.
22. Jiang, L.; Yuksel, M.; Kalyanaraman, S. Explicit rate multicast congestion control. *Comput. Netw.* **2006**, *50*, 2614–2640. [CrossRef]
23. Rizzo, L. pgmcc: A tcp-friendly single-rate multicast congestion control scheme. *ACM SIGCOMM Comput. Commun. Rev.* **2000**, *30*, 17–28. [CrossRef]
24. Widmer, J.; Handley, M. TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification. IETF RFC 4654. Available online: <https://datatracker.ietf.org/doc/rfc4654/> (accessed on 2 June 2021).
25. Handley, M.; Floyd, S.; Padhye, J.; Widmer, J. TCP Friendly Rate Control (TFRC): Protocol Specification. IETF RFC 5348. Available online: <https://datatracker.ietf.org/doc/html/rfc5348> (accessed on 2 February 2021).
26. Bouras, C.; Gkamas, A.; Kioumourtzis, G. Adaptive smooth multicast protocol for multimedia transmission: Implementation details and performance evaluation. *Int. J. Commun. Syst.* **2010**, *23*, 299–333. [CrossRef]
27. Li, J.; Yuksel, M.; Fan, X.; Kalyanaraman, S. Generalized multicast congestion control. *Comput. Netw. Int. J. Comput. Telecommun. Netw.* **2007**, *51*, 1421–1443. [CrossRef]
28. Jiang, L.; Kalyanaraman, S. MCA: A Rate-based End-to-end Multicast Congestion Avoidance Scheme. In Proceedings of the 2002 IEEE International Conference on Communications (ICC 2002), New York, NY, USA, 28 April–2 May 2002; pp. 2341–2347.
29. Chen, J.; Arumathurai, M.; Fu, X.; Ramakrishnan, K.K. Reliable Publish/Subscribe in Content-Centric Networks. In Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, Hong Kong, China, 12–16 August 2013; pp. 21–26.
30. Stais, C.; Xylomenos, G.; Voulimeneas, A. A reliable multicast transport protocol for information-centric networks. *J. Netw. Comput. Appl.* **2015**, *50*, 92–100. [CrossRef]
31. Su, K.; Ramakrishnan, K.K.; Raychaudhuri, D. Scalable, network-assisted congestion control for the MobilityFirst future internet architecture. In Proceedings of the 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Rome, Italy, 13–15 June 2016; pp. 1–2.
32. Nour, B.; Mastorakis, S.; Ullah, R.; Stergiou, N. Information-Centric Networking in Wireless Environments: Security Risks and Challenges. *IEEE Wirel. Commun.* **2021**, *28*, 121–127. [CrossRef]
33. Khelifi, H.; Luo, S.; Nour, B.; Mounghla, H. LQCC: A Link Quality-based Congestion Control Scheme in Named Data Networks. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–6.
34. Nguyen, D.; Fukushima, M.; Sugiyama, K.; Tagami, A. Efficient multipath forwarding and congestion control without route-labeling in CCN. In Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW), London, UK, 8–12 June 2015; pp. 1533–1538.
35. Nour, B.; Khelifi, H.; Hussain, R.; Mounghla, H.; Bouk, S.H. A Collaborative Multi-Metric Interface Ranking Scheme for Named Data Networks. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 2088–2093.
36. Zeng, L.; Ni, H.; Han, R. An Incrementally Deployable IP-Compatible-Information-Centric Networking Hierarchical Cache System. *Appl. Sci.* **2020**, *10*, 6228. [CrossRef]

37. Dukkupati, N.; Kobayashi, M.; Rui, Z.S.; McKeown, N. Processor sharing flows in the Internet. *Lect. Notes Comput. Sci.* **2005**, *3552*, 271–285.
38. Badov, M.; Seetharam, A.; Kurose, J.; Firoiu, V.; Nanda, S. Congestion-aware caching and search in information-centric networks. In Proceedings of the 1st ACM Conference on Information-Centric Networking, Paris, France, 24–26 September 2014; pp. 37–46.
39. Papageorgiou, M.; Hadj-Salem, H.; Blosseville, J.M. ALINEA: A local feedback control law for on-ramp metering. *Transp. Res. Rec. J. Transp. Res. Board* **1991**, *1320*, 58–64.