electronics

Article

A Link Status-Based Multipath Scheduling Scheme on Network Nodes

Hongyu Liu [1,2], Hong Ni [1,2] and Rui Han [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; liuhy@dsp.ac.cn (H.L.); nih@dsp.ac.cn (H.N.)
[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: hanr@dsp.ac.cn; Tel.: +86-138-1107-7380

Abstract: Traditional internet protocol (IP) networks, adhering to a "best-effort" service model, typically utilize shortest-path routing for data transmission. Nevertheless, this methodology encounters limitations, especially considering the increasing demands for both high reliability and high bandwidth. These demands reveal shortcomings in this routing strategy, notably its inefficient bandwidth utilization and fault recovery capabilities. The method of multipath transmission has been extensively researched as a solution to these challenges. With the emergence of innovative Internet architectures, notably information-centric networking (ICN), network nodes have gained enhanced capabilities, opening new avenues for multipath transmission design. This paper introduces a multipath scheduling approach for network nodes, capitalizing on the advanced features of these modern nodes. It reimagines the conventional next-hop node as a group of potential next-hop nodes based on both global and local routing strategies and assigns traffic shares to each node within this group for balanced traffic distribution. Network nodes are configured to periodically review and adjust traffic shares according to the link statuses. If scheduling cannot be completed within the set, feedback is sent to upstream nodes. Simulations demonstrate that this approach effectively leverages network path variety, improves bandwidth usage and throughput, and minimizes average data transmission time.

Keywords: multipath transmission; multipath scheduling; network transmission; routing forwarding

check for updates

Citation: Liu, H.; Ni, H.; Han, R. A Link Status-Based Multipath Scheduling Scheme on Network Nodes. *Electronics* **2024**, *13*, 608. https://doi.org/10.3390/electronics13030608

Academic Editor: Christos J. Bouras

Received: 10 January 2024
Revised: 26 January 2024
Accepted: 31 January 2024
Published: 1 February 2024

Copyright: © 2024 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

When the processing capacity of nodes in a network is insufficient to handle simultaneous information processing, it can lead to congestion, impacting the network's overall throughput [1]. It is important to note that a network's actual throughput is not solely determined by its theoretical capacity. Routing strategies play a crucial role as well [2–5]. For instance, having many overlapping transmission paths in a network increases the load on the nodes at these intersections. These nodes can become choke points, limiting the network's throughput. Meanwhile, the network's redundant links may remain underutilized. For instance, the common shortest-path routing method used in traditional networks, which directs data packets through the shortest route based on network topology, is simple and effective in preventing loops. Nevertheless, this approach does not account for the load on nodes that the data packets traverse, resulting in a concentration of packets at highly connected nodes, creating congestion. Conversely, completely avoiding path overlap can lead to underutilization of network resources on each link, which wastes resources, increases transmission delays, and degrades user experience. Therefore, selecting the right paths for data transmission is crucial, and this has been a focus of extensive research alongside the development of the Internet.

To resolve these challenges, the concept of multipath transmission has been thoroughly researched. This approach involves using multiple paths for data transmission instead of relying solely on the shortest path. It helps to alleviate the strain on bottleneck nodes caused by traffic convergence, thereby improving throughput and enhancing the robustness of the transmission. However, in traditional network designs where nodes are mainly responsible for routing data and the sender controls the transmission, current multipath solutions often face problems such as high complexity, limited flexibility, and narrow applicability.

To effectively address the limitations inherent in traditional Internet architectures, various advanced network architectures have been proposed by researchers. Among these, information-centric networking (ICN) is particularly outstanding [6]. This includes approaches such as content-centric networking/named data networking (CCN/NDN) [7,8] and IP-compatible networks such as SEANet [9]. These architectures give special consideration to the role of network nodes, enhancing them with innovative functionalities. For instance, they adopt a control-forward separation model akin to software-defined networking (SDN) [10] on network nodes, which significantly boosts the computational power of network nodes. Additionally, these nodes are offered with cache space, enabling them to store and fulfill content requests. These new features enable network nodes to not only relay data but also manage and oversee data transmission processes.

Our objective is to harness these enhanced network nodes for multipath scheduling. This involves using the variety of links in the network to circumvent localized traffic congestion. The aim of this study is to elevate transmission rates, improve overall network bandwidth utilization, and enhance throughput, all within the limitations of existing network resources. We propose a multipath scheduling scheme based on link status for network nodes. In this scheme, nodes create sets of next-hop scheduling alternatives to conventional next-hop nodes, using both global and local routing information. They then periodically refresh link statuses and modify the traffic distribution for each next-hop node in the set based on these link statuses. When scheduling cannot be completed within the existing set, feedback is relayed to upstream nodes. Our algorithm is versatile and applicable for both comprehensive network-wide deployment of network nodes and incremental deployment through overlay. Moreover, this paper presents a specific design of incremental deployment using overlay. The key contributions of this paper include the following:

- A novel scheduling algorithm for network nodes utilizing multiple paths is proposed. The network node reimagines the conventional next-hop node as a group of potential next-hop nodes to improve forwarding ability, considering global and local routing strategies. Then, the network node regularly modifies the traffic distribution for each next-hop node within the group based on the corresponding link status to reduce link congestion.
- A multipath scheduling feedback mechanism is devised for network nodes. If a node is unable to alleviate link congestion through the proposed scheduling algorithm, it is equipped to inform upstream nodes actively. Then, the upstream nodes assume the responsibility for scheduling to reduce node congestion.
- A specific design based on network nodes using the control and forwarding separation mode is proposed. The control plane, similar to SDN, runs the proposed multipath scheduling algorithm and feedback mechanism. In the data plane, we designed the packet header and forwarding flow tables, ensuring streamlined functionality. The design ensures that the proposed scheduling scheme can be effectively put into practice.

This paper is structured as follows: Section 2 reviews prior research on multipath methodologies. Section 3 presents the problem statement. Section 4 elaborates on the newly proposed multipath scheduling algorithm and the feedback system for network nodes. Section 5 explores the practical design of this algorithm and feedback mechanism, focusing on link-status information. Section 6 covers the analysis of experimental results. Section 7 engages in extended discussions. The paper concludes with a summary in Section 8.

2. Related Work

Multipath transmission has been widely studied for its ability to improve transmission efficiency and robustness. According to different scheduling decision makers, existing strategies for multipath scheduling are generally categorized into two types.

One type is scheduling based on the sender in the transport layer. Typical examples include MPTCP [11], SCTP [12], and MPQUIC [13,14]. MPTCP divides a single connection's data flow into multiple TCP subflows, each of which can choose a different path for data transmission. Although MPTCP can aggregate link bandwidth and improve transmission throughput, it faces issues like queue head blocking and packet disorder. SCTP was originally a standardized signaling protocol developed by the IETF working group and later developed into a universal protocol for the transport layer. Like TCP, it is integrated into the operating system's kernel and can provide reliable and efficient data transmission services for both ends of the transmission. In addition, SCTP adopts the reliable transmission and congestion control mechanisms of TCP but also offers corresponding improvements. MPQUIC is a multipath scheme proposed in recent years, which is based on QUIC and utilizes multiple paths for data transmission. Reference [15] proposes a path management method based on retransmission, which adopts the design concept of using only one optimal path for a connection and other paths as backups to ensure that data packets can reach the receiver in order. For short-stream data transmission, reference [16] proposes a novel MPTCP path management method, which temporarily closes slower transmission paths based on the delay difference between paths, allowing short-stream data packets to be transmitted from the best quality path, thereby reducing the competition time of short streams. The authors of reference [17] designed a quality-aware path selection mechanism by evaluating transmission delay and bandwidth information. The authors of reference [18] developed a path management algorithm based on machine learning, which collects quality parameters such as transmission rate, throughput, and round-trip delay through calculation and evaluates path quality based on different feature values. The authors of reference [19] proposed source routing technology. The source router encodes the network path information in the data packet's header, making the transmission path definable. However, source routing requires the source node to predict the topology and status information of the entire network, and routers in the path need to correctly process source routing packets, which leads to high network management complexity [20] and poor security [21], limiting the further development of source routing.

Another type is scheduling based on the network node in the network layer. The weight adjustment algorithm (WA) proposed in reference [22] mainly consists of weight adjustment and load optimization. Weight adjustment selects the links with the maximum and minimum loads and adjusts their weight values. Load optimization recalculates the network's load situation after adjusting the weights and leaving weight settings that improve the network's load situation. The authors of reference [23] proposed two heuristic algorithms for optimizing weights. One type is a noniterative algorithm. To formally model the original problem, the algorithm finds the dual problem of the original problem and solves it. Another approach is to use column generation algorithms to solve linear programming problems. The method used in reference [24] is similar to that reported in reference [22], which heuristically adjusts link weight settings based on the current network status to maximize the remaining network bandwidth. The authors of reference [25] designed a new routing mechanism, which selects a subset from multiple adjacent sets of routers in the current router as the next hop for traffic routing. The theoretical approximate optimal solution is achieved by simulating the unequal distribution of traffic on multiple paths. The authors of reference [26] broke the rule limitations of ECMP and proposed a weighted ECMP scheme. They then designed a heuristic algorithm based on simulated annealing to calculate an approximately optimal diversion and achieve the minimum total delay of end-to-end traffic. The authors of reference [27] proposed an effective path strategy (EPS). The authors of reference [28] proposed a new routing strategy based on the method reported in reference [27]. Two adjustable parameters have been introduced. The authors

or reference [29] improved the effective path-routing process proposed in reference [27]. The authors of reference [30] proposed a new hybrid routing strategy that calculates the probability of information packets being transmitted to neighboring nodes based on the degree of the node and a probability formula. This strategy considers both the node's ability to process packets and the queue length at the node. The authors of references [31–33] proposed a hybrid routing strategy called the Traffic Awareness Protocol (TAP). In this strategy, the congestion situation of the network is perceived by the queue length of nodes, and corresponding measures are taken to disperse information packets and improve the transmission efficiency of information packets.

This paper introduces a multipath scheduling approach for network nodes, capitalizing on the advanced features of these modern nodes. It reimagines the conventional next-hop node as a group of potential next-hop nodes based on both global and local routing strategies and assigns traffic shares to each node within this group for balanced traffic distribution. Network nodes are configured to periodically review and adjust traffic shares according to the link statuses. If scheduling cannot be completed within the set, feedback is sent to upstream nodes. Compared with scheme scheduling based on the sender in the transport layer, this scheme is not limited by the shortest path routing or the number of user addresses, making it a more flexible and high-throughput solution. Compared with existing scheme scheduling based on the network node in the network layer, this scheme can adapt to link status changes and modify the traffic distribution for each next-hop node in the set based on these link statuses. It also adds a feedback mechanism, which can bypass local traffic hot spots; effectively leverages network path variety; and improves bandwidth usage and throughput.

3. Problem Statement

Utilizing multipath transmission is beneficial for maximizing the diversity of network paths, thereby enhancing network bandwidth utilization and overall throughput. Existing strategies for multipath scheduling are generally categorized into two types: scheduling based on the sender in the transport layer and scheduling based on the network node in the network layer. Both methods have their respective limitations.

3.1. Sender-Based Scheduling

In scheduling schemes centered on the sender, multiple paths are created by utilizing the various addresses linked to the sender and receiver. This method allows the sender to collect data such as the round-trip time (RTT) and bandwidth of each path for better scheduling decisions. However, a key limitation of this strategy is its reliance on having multiple addresses available at both the sender and receiver, which is not always the case. Moreover, senders often face challenges in fully understanding the complexities of network links. As a result, the number of viable transmission paths is generally quite restricted, leading to an underutilization of the diverse network links. Furthermore, each chosen path is typically the shortest, which can cause issues such as traffic clustering, the formation of local hot spots, and competition over shared network bottlenecks.

3.2. Network Node-Based Scheduling

Network node-based scheduling approaches leverage the global and local routing data available at network nodes. This enables the utilization of a wider array of transmission paths than is possible with sender-based scheduling. Nevertheless, network nodes usually have to rely solely on local link data for their scheduling choices. Bound by the traditional network structure, where nodes are mainly in charge of data forwarding and senders handle control, the existing network node-based scheduling methods are often quite basic. For instance, Equal-Cost Multi-Path (ECMP) primarily utilizes local link data for scheduling and faces several challenges. First, a critical issue is its inability to accurately perceive bottlenecks along the entire path. When upstream nodes allocate high bandwidth, the downstream nodes can be overwhelmed, causing congestion. Secondly, nodes often only

assess congestion at the immediate next-hop node, redistributing incoming traffic to various downstream nodes to prevent overload on a single path. However, congestion can still occur if multiple upstream nodes direct their traffic to a single downstream node. Even if individual links are not congested, this can exceed the receiving node's scheduling capacity, leading to congestion.

4. Proposed Multipath Scheduling Algorithm and Feedback Mechanism

This section details the proposed multipath scheduling algorithm and feedback mechanism on network nodes. Based on the previous analysis, in this scheme, nodes create sets of next-hop scheduling alternatives to conventional next-hop nodes, using both global and local routing information. They then periodically refresh link statuses and modify the traffic distribution for each next-hop node in the set based on these link statuses. The network node monitors and maintains the status of adjacent links and decides if scheduling adjustments are necessary. When scheduling cannot be completed within the existing set, feedback is relayed to upstream nodes. There are multiple options for global routing strategies, and this article introduces the most widely applicable shortest path as an example. For ease of understanding, this article distinguishes between links and paths. The two adjacent nodes are called links, connected in series to form a path. Network nodes choose different next-hop nodes, which means they choose different links, then choose different paths.

4.1. Next-Hop Set

We reimagine the conventional next-hop node as a set of potential next-hop nodes. Each node in the set corresponds to a link. If each network node schedules the traffic of congested links to any other link, it will form many loops, which is unacceptable. Therefore, we need to avoid forming loops.

First, we initialize the node set to include all surrounding nodes; then, based on global routing information, which is the result of the shortest path routing, we remove the node from the set where the next hop is the current node. As shown in Figure 1, R1 receives routing information from User1, R3, and R2; deletes User1, with the next hop being the current node; and combines R2 and R3 into the next-hop set. Similarly, R2 combines R1 and R4 to form the next-hop set. Due to the expansion of network node forwarding from a single next-hop node to multiple next-hop nodes, a loop appears between nodes R1 and R2.

To avoid loops, only one node can retain the link between R1 and R2, and the other should not use this link for forwarding. However, we cannot determine this in advance based on routing information because which node should retain this link is determined by the direction of the data flow. If User1 sends the data, R1 should retain this link. If User2 sends the data, R2 should retain this link. Therefore, we designed a random shield mechanism. When a packet is found to be forwarded from the incoming port, it indicates that a loop has occurred. We then find the corresponding set of next-hop nodes based on the packet routing and shield the next-hop node corresponding to the port it is forwarded to. We just shield a random amount of time instead of deleting because, at this point, it may be possible for the link between nodes R1 and R2 to be shielded by both nodes. This avoids loops but reduces the utilization of the path. Network nodes do not forward data to the shielded next-hop node, and the shield time is an integer multiple of the control period (T). The control period is explained in detail in the following text. If no packets enter this node from the corresponding link after the shield time, the shield is unbanned. Otherwise, a longer random shield time continues to be set. Without loss of generality, we assume that User1 sends data to User3. R2 detects packets from R1 to IPa, thus shielding the R1 node in the next-hop set for a while.

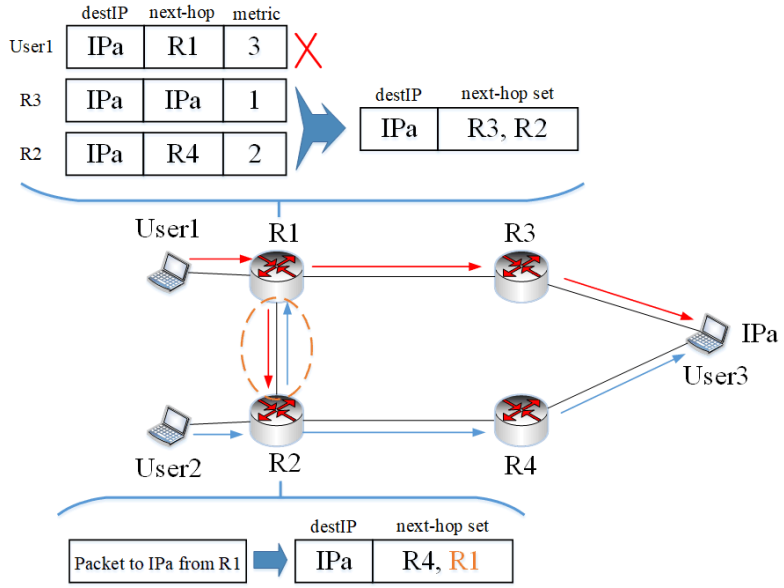


Figure 1. Loop avoidance.

4.2. Scheduling Algorithm

After establishing the next-hop set, assigning traffic shares to each node within this set for balanced traffic distribution is necessary. Existing scheduling schemes, such as random and polling schemes, are simple and easy to implement, but they cannot adapt to changes in network status and achieve unstable performance. To adapt to changes in network status, some schemes design probability functions based on queue length. Ports with longer queues have a lower forwarding probability, while ports with shorter ones have a higher forwarding probability. This method primarily utilizes local link data for scheduling and faces several challenges. First, a critical issue is its inability to accurately perceive bottlenecks along the entire path. When upstream nodes allocate high bandwidth, the downstream nodes can be overwhelmed, causing congestion. Secondly, nodes often only assess congestion at the immediate next-hop node, redistributing incoming traffic to various downstream nodes to prevent overload on a single path. However, congestion can still occur if multiple upstream nodes direct their traffic to a single downstream node. Even if individual links are not congested, this can exceed the receiving node's scheduling capacity, leading to congestion.

4.2.1. Link Metrics

Our scheme first considers link metrics. Link metrics are one of the critical factors affecting the path selection of network nodes. For example, in the widely used shortest-path routing method, nodes can obtain global topology information, then use the number of hops as a metric and choose the path with the lowest metric value. In the scheme proposed in this article, network nodes calculate link metrics based on relatively stable factors, such as bandwidth, latency, and hop count, to represent the performance characteristics of the link. We prioritize selecting links with smaller metric values.

$$Metric = a \times W_l + b \div B_s + c \times D_l \quad (1)$$

where W_l represents the minimum number of hops required to reach the destination after selecting the link, which can be obtained from the global shortest path routing results; B_s is the bandwidth of the link; and D_l is the delay of the link, which can be obtained by node detection. Among them, a, b, and c are constant coefficients that can control the degree of influence of different parameters on link metrics. Our main goal is to utilize path diversity based on the shortest path so that nodes can support higher transmission rates. Therefore, in this article, we set the constant coefficients to $a = 100$, $b = 10$, and $c = 1$. This means

we first prioritize links with lower numbers of forwarding hops. When the number of forwarding hops is similar, we prioritize links with larger bandwidth. If the bandwidth is also identical, we prioritize links with lower latency. It is worth noting that, on the one hand, the factors considered in the link metrics mentioned in this article are mainly the relatively stable characteristics of the links, which do not change frequently. On the other hand, the surrounding link metrics recorded by each node in this article are only used for the current node's scheduling decisions, without informing other nodes in the network.

4.2.2. Link Status

The link scheduling algorithm proposed in this article assigns traffic shares to each node within this set for balanced traffic distribution. There are two allocation methods for the initial status; one is to set the node corresponding to the smallest metric link to 100% and the other nodes to 0. Another approach is to allocate traffic proportionally based on link bandwidth. This article defaults to using the first method for initialization. After initialization, the link scheduling algorithm proposed in this article maintains and updates the status of links corresponding to next-hop nodes in the set according to the control cycle (T) and adjusts the traffic ratio of each next-hop node according to the link status. The longer the length of the port queue, the longer the control period (T). According to simulation results, under the conditions investigated in this article, $T = 1$ s has the best scheduling effect. Therefore, under the default condition investigated in this article, T is set to 1 s.

The detailed steps of the link scheduling algorithm are described in Algorithm 1. Input B_t and F_s represent the total bandwidth of the link and the amount of forwarding in the link, respectively. These pieces of information are collected and detected by network nodes. The output is V_c, which is the value at which the next-hop set cannot complete the scheduling.

The path scheduling algorithm proposed in this article is divided into two parts. The first part is from line 1 to line 10, which determines whether the link needs to be scheduled based on link bandwidth and forwarding volume and calculates how much traffic needs to be scheduled. We initialize the link status ($status$) to a saturated status with a scheduling value ($status_value$) of 0 (line 2). This status link does not require scheduling and cannot receive traffic from other links. When the forwarding volume exceeds or is equal to the bandwidth, the link is congested, and the port queue continues to grow. We calculate the proportion of traffic that needs to be scheduled to transition the link to a saturated status (lines 3–5). When the forwarding volume is lower than the bandwidth, it is unsaturated. At this time, the link bandwidth utilization is low, and it can receive traffic scheduled by other links. We also calculate how much traffic can be scheduled when it reaches saturation status (lines 6–9). To avoid unnecessary traffic scheduling caused by jitters between congested and unsaturated status, we set a constant coefficient (α, where $0 < \alpha < 1$). If α is too small, it affects the scheduling effect, while if α is too large, it may increase jitter. In this article, α is taken as 0.9.

The network node updates the link status once every control cycle (T). When a congested link occurs, the flow proportion of the next-hop node set containing the congested link is adjusted. In the second part of the algorithm, from line 11 to line 31, we adjust the flow allocation proportion in the next-hop set according to the link status. The demand for transmission in the network is constantly changing, and the status of network links also changes accordingly. A link may belong to multiple different next-hop sets at the same time. If link A traffic needs to decrease by 20%, we need to reduce the traffic forwarded to A by 20% among all next-hop sets containing A. The total amount of traffic that needs to be scheduled is initialized as $V_c = 0$, and the set of next-hop nodes that can receive scheduled traffic (Set_u) is empty (line 12). Then, the node in the next-hop node set is traversed. If the link status corresponding to the node is congested, the proportion that needs to be scheduled is calculated based on the current traffic allocation value ($ratio$) and the link scheduling value ($status_value$), then added to V_c (lines 13–15). If the link

status is unsaturated, the corresponding node is added to Set_u (lines 16–19). We sort the nodes in Set_u in ascending order of their corresponding link metrics (line 20). Finally, we sequentially select nodes (lines 21–22) from Set_u. We calculate the proportion of traffic that a node can receive based on its current allocation value (*ratio*) and the link scheduling value (*status_value*). We take the smaller value between the node that can receive traffic and the total scheduling value (V_c) as the current scheduling value (V_d) (line 24). Current node allocation value *ratio* plus V_d (line 25), the total scheduling value V_c minus V_d (line 26).

Algorithm 1: Multipath scheduling algorithm

Input: B_t, F_s
Output: V_c
1 **Function** `GetLinkStatus(` B_t, F_s`):`
2 Initialization: $status = saturation, status_value = 0\%$;
3 **if** $F_s > B_t$ **then**
4 $status = congestion$;
5 $status_value = (F_s - B_t)/F_s$;
6 **else if** $F_s < \alpha B_t$ **then**
7 $status = unsaturation$;
8 $status_value = (\alpha B_t - F_s)/F_s$;
9 **end**
10 **return** *status*,*status_value*
11 **Function** `Schedule(` *next_set*`):`
12 Initialization: $V_c = 0, Set_u$;
13 **for** $node_m \in next_set$ **do**
14 **if** $node_m.link_m.status == congestion$ **then**
15 $V_c += node_m.link_m.status_value \times node_m.link_m.ratio$;
16 **else if** $node_m.link_m.status == unsaturation$ **then**
17 $Set_u.add(node_m)$;
18 **end**
19 **end**
20 $sort(Set_u, link\ metric, ascending)$;
21 **while** $V_c > 0$ **do**
22 $node_k = Set_u.top()$;
23 **if** $node_k != null$ **then**
24 $V_d = min(V_c, node_k.link_k.ratio \times node_k.link_k.status_value)$;
25 $node_k.link_k.ratio += V_d$;
26 $V_c -= V_d$;
27 **else**
28 *break*;
29 **end**
30 $Set_u.pop()$;
31 **end**
32 **return** V_c

4.3. Feedback Mechanism

Congestion can still occur if multiple upstream nodes direct their traffic to a single downstream node. Even if individual links are not congested, this can exceed the receiving node's scheduling capacity, leading to congestion. Therefore, we designed a feedback scheduling mechanism.

Figure 2 shows the process of the feedback scheduling mechanism, where we first call the scheduling algorithm to calculate the scheduling value (V_c) that requires feedback scheduling. If $V_c > 0$, we select $node_i$ from all neighboring nodes in sequence. If $node_i$ is not in the next-hop set (*next_set*), a feedback scheduling message containing the re-

quired scheduling value (V_c) is constructed and sent to $node_i$. After receiving the feedback scheduling message, $node_i$ sets the link status of the corresponding link to congested, and the scheduling value (*status_value*) takes the larger value of the current statistical value (*status_value*) and the value of V_c in the message.

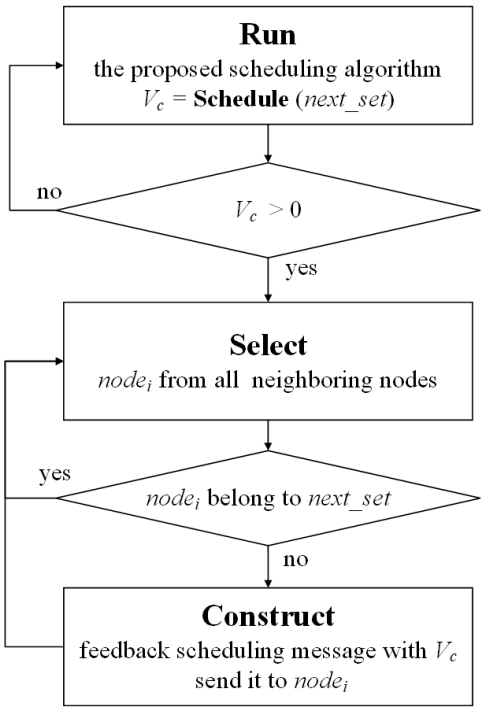


Figure 2. Feedback flow chart.

The metric and status algorithms of each link have a constant time complexity of O (1). The time complexity of the next-hop set construction algorithm is O (N), where N is the number of all nodes around the node. The time complexity of path scheduling and feedback algorithms within the set is O (M), where M is the number of nodes within the next-hop set. These algorithms are completed in the control plane of nodes and do not increase the processing complexity of packet forwarding. Scheduling feedback messages are sent once in one or several control cycles without significantly increasing the additional burden on the link.

5. Scheme Implementation and Design

In this section, we introduce the design of path-scheduling schemes based on multipath scheduling algorithms and feedback mechanisms on network nodes. In traditional network designs where nodes are mainly responsible for routing data, the sender controls the transmission, and control is coupled with forwarding on network nodes, current multipath solutions often face problems such as high complexity, limited flexibility, and narrow applicability. The solution proposed in this article adopts network nodes in a new network architecture, which have the following characteristics:

- Nodes have additional computing power, and control and forwarding are separated. The algorithm proposed in this article can perform distributed computing on local controllers on nodes without the need for centralized control planes.
- Compatible with existing IP networks, incremental deployment can be carried out in an overlay manner within the existing IP network.

5.1. Model Introduction

All nodes in the network are enhanced network nodes, which is the most ideal and simple scenario, and only the proposed scheduling algorithm needs to be run on each

node. However, the trend of network development is gradual, so we will take incremental deployment as an example to introduce our design solution.

As shown in Figure 3, in the scenario of incremental deployment, when the enhanced node runs the proposed scheduling algorithm, routing should be based on overlay topology, which means that the underlying traditional IP nodes are invisible to the algorithm. The direct link between enhanced nodes is a link composed of one or more traditional IP node links. Therefore, in the scenario of incremental deployment, nodes can obtain information such as link bandwidth and latency required by scheduling algorithms through network measurements. There has been extensive research on network measurement [34], which can be divided into active, passive, and hybrid measurements [35]. For example, the scheme based on alternative marking performance measurement (AM-PM) can measure latency per hop [36,37], while the passive measurement scheme based on group dispersion and autocorrelation can measure the available bandwidth of the link [38].

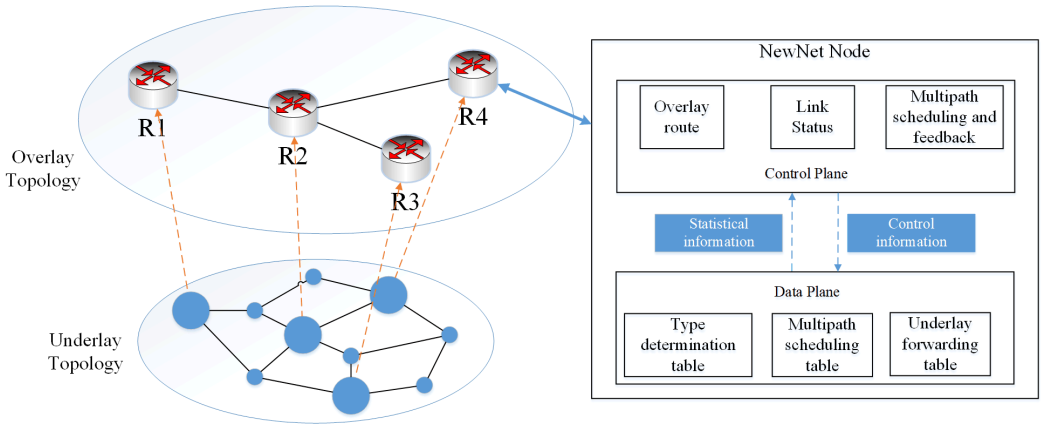


Figure 3. System model.

All enhanced nodes adopt a design pattern that separates control and forwarding. In the data plane, packets are forwarded according to the flow table, and statistical information and scheduling feedback messages are sent to the control plane. Scheduling algorithms are run in the control plane to make scheduling decisions. When scheduling decisions change, the flow table is notified to adjust and achieve transmission scheduling. When a scheduling feedback message needs to be sent, the control plane sends it to the data plane.

5.2. Packet Header Design

The design of the multipath transmission packet is shown in Figure 4. In order to ensure compatibility with existing IP networks, the proposed multipath transmission packet header is located in the payload part of the IP packet and divided into three fields.

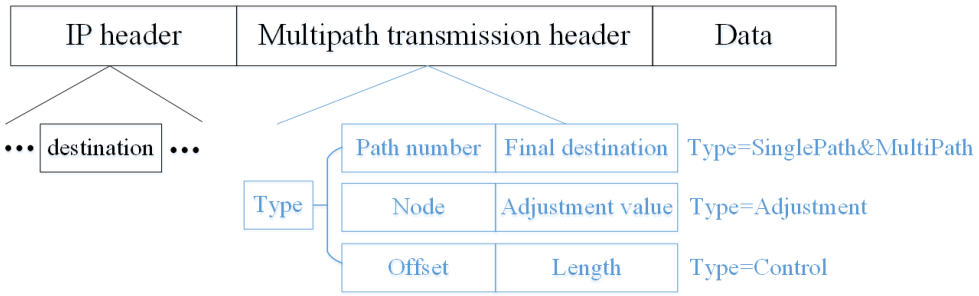


Figure 4. Multipath transmission packet header.

The type field represents the transmission type, *Singlepath* represents single-path transmission, *Multipath* represents multipath transmission, *Adjustment* represents a scheduling feedback message, and *Control* represents a control message.

When the type is *Singlepath* or *Multipath*, the following two fields represent the path number and final destination, respectively. The former represents the upper limit of the transmission path, while the latter represents the final destination address.The more paths, the higher the bandwidth and the faster the transmission rate. However, the more paths there are, the greater the differences between them, especially in terms of latency, which leads to severe packet disorder. Much research on out-of-order reordering at the transport layer allows us to reorder out-of-order packets at the receiving end. However, the more disorder, the greater the time and space resources required for reordering and even the problem of blocking the queue head. Therefore, we designed the path-number field to limit the number of paths, and the transmission can balance the actual bandwidth and rearrangement cost to determine the value of the path number. We need to modify the destination address field of the IP header so that the intermediate node can forward the data packet to the next-hop enhanced node in the overlay topology according to our schedule. Therefore, it is necessary to record the final destination address.

When the type is *Adjustment*, the following two fields represent node and adjustment value, respectively. The former is the identification of the node that sends scheduling feedback messages and can use the IP of its sending port. The latter is the value that the node needs to schedule.

When the type is *Control*, the following two fields represent offset and length, respectively. The former is the starting offset that needs to be modified in the multipath scheduling table state area. The latter refers to the length of the modified content.

5.3. Flow Table Design

In response to the growing demand for traffic in networks, researchers have launched many innovative technologies, including multipath approaches, that can meet requirements such as high throughput and low latency. Network infrastructure needs to be significantly improved to support these emerging technologies, and this improvement process requires addressing challenges such as network complexity, maintenance costs, and the installation environment. Therefore, to facilitate network infrastructure improvement, network softwarization technology [39,40], including SDN and Programmable Data Plane (PDP), has received attention. SDN can reduce deployment costs, but centralized architecture has poor scalability and limited data plane functionality. Another form of network softwarization, PDP [41,42], allows for customized packet operations and access to device-internal states, allowing for programmable processing of packets on the data plane, thereby reducing dependence on the control plane. The P4 protocol [43] and the POF protocol [44] are both protocol-independent forwarding methods that support data plane programmability.The POF protocol is simpler and easier to understand. Therefore, we use a switch based on the POF protocol as the data plane [45]. This type of switch receives and installs flow tables distributed by the control plane through the POF protocol, then processes data packets in a match–action pipeline mode. The flow table contains matching fields and action instruction blocks. In addition, POF switches can allocate state areas for packets, flows, and flow tables, achieving state programmability. Figure 5 shows the main design of our data plane pipeline, including the matching fields, actions, and state areas of the flow tables. We record the port at which the packet enters the network node in the packet state area and write it when the packet enters the node. In the multipath table state area, we record the node (*NodeIP*) and allocation (*Ratio*) values in the next-hop set corresponding to the route and record different next-hop sets from different offsets.

The type judgment table is responsible for determining the transmission type. If the type is *Control*, it indicates that it is a control message from the control plane. The records in the multipath table state area are modified according to the specified offset, length, and content in the message. Suppose the transmission message type is *Adjustment*. In that case, it indicates scheduling feedback messages sent by the surrounding nodes. It should be forwarded to the control plane, which adjusts the status of the corresponding link based on the message. If the type is *Multipath*, it indicates that the multipath transmission packet

is forwarded to the multipath scheduling table for processing. Other messages are directly sent to the forwarding table for processing.

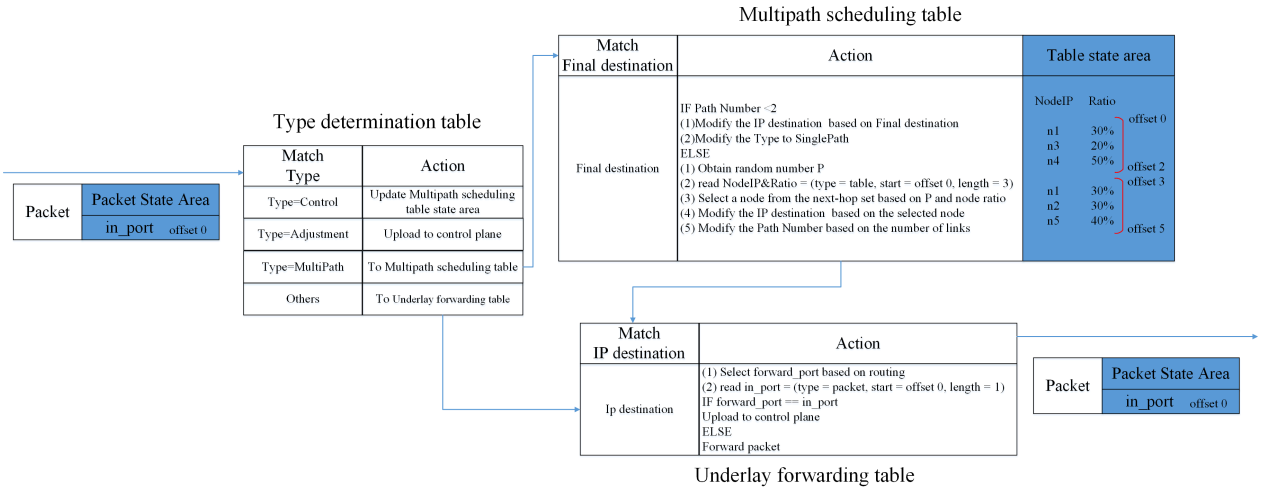


Figure 5. Multipath transmission flow table.

The multipath scheduling table is responsible for conducting multipath scheduling. First, we check if the path number is less than 2. If it is, it indicates that multipath transmission is no longer necessary. Then, we modify the destination address in the IP header to the final address in the multipath header and change the type in the multipath header to *SinglePath*. Otherwise, we obtain a random number, select a node based on the the random number and the allocation value (*Ratio*) of each node in the next-hop set, then modify the destination address in the IP header to the address of the selected next-hop node, and, finally, modify the path number in the multipath header.

$$Path\ number = Path\ number \div Links\ number \tag{2}$$

where links number represents the number of links corresponding to the nodes in the next-hop set.

The forwarding table forwards packets based on the underlying routing. The network node records the port at which the packet enters the node in the packet state area. If the forwarding and entry ports are the same, it notifies the control plane to shield the corresponding node in the next-hop set.

5.4. Control Plane

We chose a controller built as part of open-source project ONOS as the control plane [46] and designed a control application using the northbound interface provided by ONOS to implement the proposed scheduling algorithm. The controller can run standard routing protocols such as OSPF and RIP and master the topology connection relationships of network nodes.

The workflow of the control application is shown in Figure 6. Our control application first uses the set construction method introduced in Section 4, selects surrounding network nodes to form the next-hop set based on the global routing information and topological connection relationships provided by the controller, and sets the initial allocation value (*Ratio*) for the nodes in the set according to the metric of the corresponding link. Then, based on the next-hop set, a multipath scheduling table is established, along with the corresponding type determination table and underlay forwarding table. The matching fields, instruction blocks, and state areas of the flow tables are sequentially distributed to the data plane through the POF protocol.

The control application periodically reads statistical information, such as the forwarding volume of the data plane, and receives messages uploaded by the data plane. First,

based on the statistical information, including the forwarding volume and bandwidth of each link, we calculate the link status and scheduling values. If the feedback scheduling message uploaded from the data plane is received, we set the state of the corresponding link to congested. The scheduling value takes the larger value between the current calculated scheduling value and the scheduling value in the feedback scheduling message to complete the scheduling requirements of the node sending the feedback scheduling message. If a data packet is received from the data plane, it indicates that a loop has occurred. The next-hop set is determined based on the final destination in the packet; then, the next-hop node in the set is determined based on the IP destination address and shielded for a while.

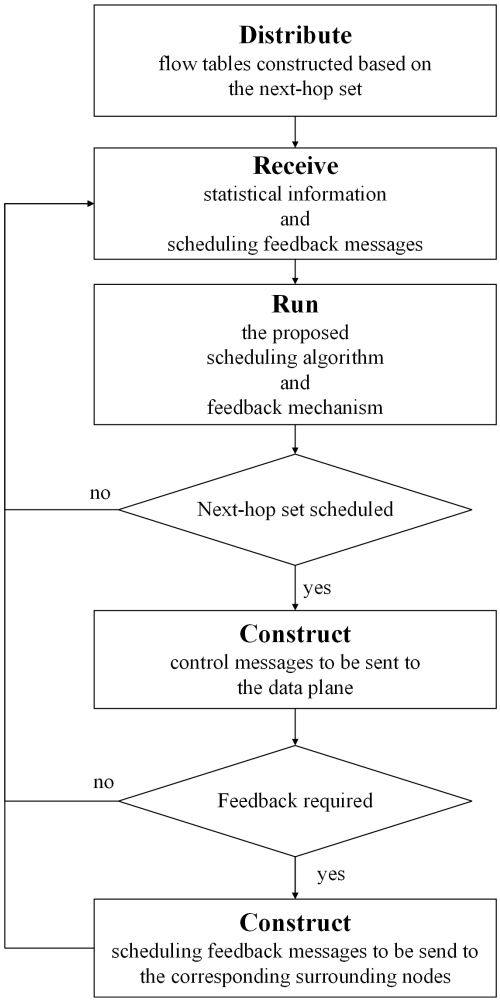


Figure 6. Control plane flow chart.

The control application executes the multipath scheduling algorithm introduced in Section 4 based on the link status and scheduling values. In the next-hop set, the traffic of nodes corresponding to congested links and blocked nodes is allocated to the nodes with a link state of unsaturated in the set in ascending order of link metric to update the allocation ratio of each node. The control messages are constructed through the POF protocol and sent to the data plane, including the starting offset and length of the scheduled next-hop set in the state area and the allocation ratio of each node. The data plane modifies the allocation ratio in the state area of the multipath scheduling table based on the control message and forwards packets according to the new allocation ratio, achieving adjustment of traffic distribution.

After completing the multipath scheduling algorithm, if feedback is required, the feedback scheduling mechanism introduced in Section 4 is executed. Surrounding nodes that do not belong to the next-hop set are selected in sequence, and feedback scheduling

messages are constructed and sent to the data plane, including the identification of the sending node and the required scheduling value. The data plane forwards feedback scheduling messages to the corresponding network nodes. The feedback scheduling message can be transmitted using protocols such as UDP, and the required fields for the feedback scheduling message need to be included.

The control application enters the next control cycle, continues to read statistical information from the data plane, receives messages uploaded from the data plane, updates link status, executes multipath scheduling algorithms and feedback scheduling mechanisms, and sends control and feedback scheduling messages to the data plane.

This section introduces the design of the proposed network node multipath scheduling scheme in the overlay incremental deployment scenario. The direct link between enhanced network nodes is a link composed of one or more traditional IP nodes. Therefore, the data plane selects the next-hop set based on the final destination in the packet and modifies the IP destination address in the packet to the IP address of the selected next-hop node to ensure that the traditional IP nodes in the middle can correctly forward the packet to the selected next-hop enhanced node. In the scenario of full network deployment, the adjacent enhanced network nodes are directly connected, so the data plane selects the next-hop set based on the IP destination address in the packet and modifies the MAC destination address in the packet to the MAC address of the selected next-hop node. In addition to network nodes, the proposed solution also requires devices such as terminals and gateways to support the fields required for multipath transmission in packets.

6. Results

The multipath scheduling and feedback algorithm based on the link status on network nodes and the corresponding scheme design has been elucidated. Next, we introduce the experimental process in this section. First, we introduce the experimental parameters and present and analyze the experimental results.

6.1. Experimental Setup

We chose the NS-3 (Version 3.34) simulation platform based on C++ implementation for the experiment. NS-3 is an easy-to-use discrete event network simulator. This simulation experiment mainly includes three types of nodes: the server node that sends data, the user node that receives data, and the router node that forwards data. The proposed scheme is mainly applied within autonomous systems. To simulate and test the performance of the proposed solution in the network, we used 27 nodes to construct the network shown in Figure 7. It includes 22 router nodes, 1 server node, and 4 user nodes.

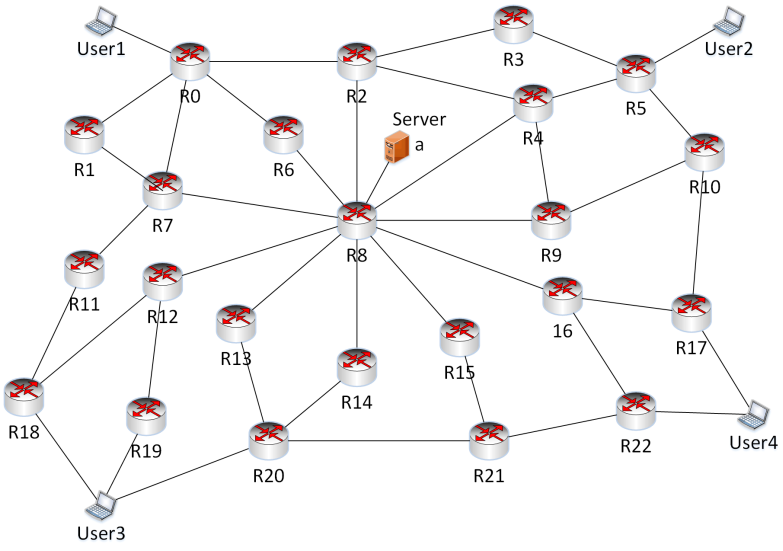


Figure 7. Experimental topology.

In the simulation experimental network shown in Figure 7, all router nodes are enhanced nodes with multipath scheduling processing capabilities, and their connections constitute the network links required for the experiment. The link between the router node and the server node forms the data sender access link required for the experiment. The link between the router node and the user node forms the data-receiving end-access link required for the experiment. To reflect the impact of different network structures on multipath scheduling strategies, we allocated a single access link for users User1 and User2 and set multiple access links for users User3 and User4. An equivalent shortest path was set between Server and User1 and User3, while there is no equivalent shortest path to User2 and User4.

We implemented the algorithms introduced in Sections 4 and 5 (represented as NODE SCHEDULING) based on the RIP routing protocol in NS-3, with a path number of 10. We compared the network node traffic scheduling scheme ECMP, multi-homing-based multipath scheduling scheme(MULT-HOMING) [47], and ECN-based dynamic weighted round-robin scheduling scheme(ECN-BASED DWRR) [48] with the proposed scheme in this paper, where ECMP adopts the built-in version in NS-3. MPTCP is a multipath scheme in the transport layer, and the sender controls its multipath scheduling. Its purpose is also to improve network throughput by utilizing multipath transmission. Therefore, we also use it as a comparison scheme and adopt the version implemented in NS-3 in article [49]. For fairness in the experiment, we utilize multi-IP transmission for all schemes when transmitting to users with multiple IPs. We conducted simulation experiments considering three aspects: bandwidth utilization, network throughput, and average transmission time. On the intermediate routing node, we chose the FIFO+DropTail queue management method with a queue length of 100 packets. The main parameter settings are shown in Table 1. When the amount of transmitted data is 0, it indicates that the size of transmitted data is not limited. Other experiments have no limitations on the data transmission size except for the experiment on transmission time.

Table 1. Experimental parameter setup.

Parameter	Value
Test Platform	NS-3
Tested Scheme	MPTCP/ECMP/MULT-HOMING/ECN-BASED DWRR/NODE SCHEDULING
Server Access Link Bandwidth/Delay	25 Mbps/10 ms
User Access Link Bandwidth/Delay	20 Mbps/10 ms
Network Link Bandwidth/Delay	1 Mbps–4 Mbps/1 ms
Queue Length	100 packets
Receive Cache Size	10 MB
Send Cache Size	10 MB
Packet Size	1052 B
Data Transferr Amount	0 MB/100 MB

6.2. Performance Evaluation

To demonstrate the performance impact of the control cycle and link selection method on the multipath scheduling scheme proposed in this paper, we instructed the Server to send data to User1. The experimental results are presented as follows.

6.2.1. Control Cycle

Figure 8 shows the variation of network throughput with the link control cycle. From the graph, it can be seen that the throughput of the proposed scheme reaches its highest point when the control cycle is 1000 ms. This is because the control cycle is small, the statistical link status time is short, and it frequently changes, resulting in unnecessary scheduling and a decrease in throughput. When the control cycle is too long, congestion in some links cannot be scheduled promptly, which can also cause a decrease in throughput.

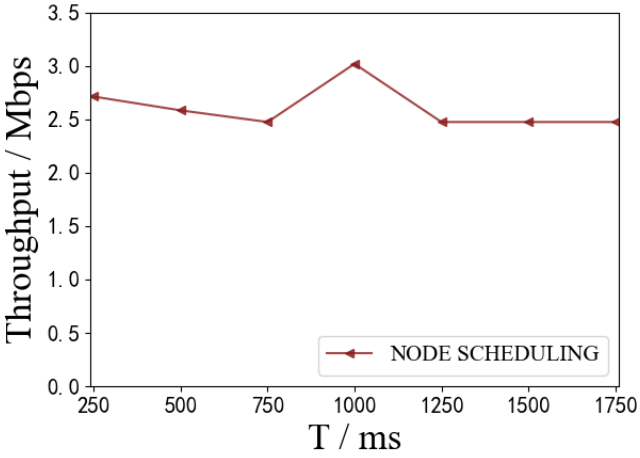


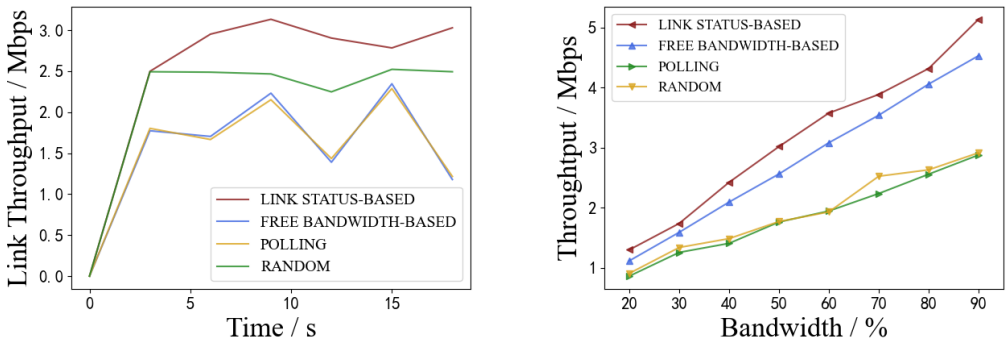
Figure 8. Control cycle.

6.2.2. Scheduling Algorithm

We chose random, polling, and free bandwidth-based scheduling algorithms to compare with the link status-based scheduling algorithm proposed in this paper. We change the network bandwidth by controlling the proportion of available bandwidth in the network.

Figure 9a shows the variation of throughput over time. During the initial transmission period, due to the slow start algorithm, the throughput of all four scheduling algorithms increases over time, and the difference is insignificant. As time goes by, the throughput of the four schemes begins to fluctuate periodically after exiting the slow start. Overall, the throughput of the scheme based on the link status proposed in this article is the highest. The throughput performance of random and polling algorithms is similar and lower than that of scheduling algorithms based on free bandwidth.

Figure 9b shows the variation of throughput with network bandwidth. As the bandwidth in the network increases, the throughput of all scheduling algorithms increases. Both random and polling algorithms distribute traffic equally across various links with equal probability, so their throughput performance is similar. The scheduling algorithm based on link-free bandwidth can allocate more traffic to links with more free bandwidth, and its throughput performance is better than that random and polling algorithms. With the increase in network bandwidth, the throughput improvement becomes more significant. At a bandwidth of 90%, its throughput is approximately 1.5 times random. The throughput of the scheme based on the link status proposed in this article is the highest, with a 13% improvement compared to scheduling algorithms based on free bandwidth when the bandwidth is 90%, which is about 1.7 times that of random selection.



(**a**) Throughput varies with time (**b**) Throughput varies with bandwidth

Figure 9. Scheduling algorithm.

6.3. Network Bandwidth

To visually demonstrate the usage of network bandwidth, we conducted comparative experiments on the performance of the five schemes in terms of path diversity and average network bandwidth utilization. Our experimental topology contains 34 links, forming many different transmission paths. We instructed the Server node to send data to four users simultaneously. The experimental results are presented as follows.

6.3.1. Path Diversity

Figure 10 shows the number of data packets passing through links under five multipath scheduling strategies. The MPTCP scheme only utilizes the shortest path between each user IP and the service node, resulting in the fewest network links that data packets pass through. ECMP improves link numbers through equivalent links. MULTI-HOMING selects paths from the shortest path of multiple IPs on a user, which is similar to the MPTCP scheme in link number. ECN-BASED DWRR is not limited to equivalent links, and its number of links can be further improved based on ECMP. The scheme proposed in this article allows for multipath scheduling on each node, making path selection more flexible than the four comparative schemes and has the highest number of links for transmission.

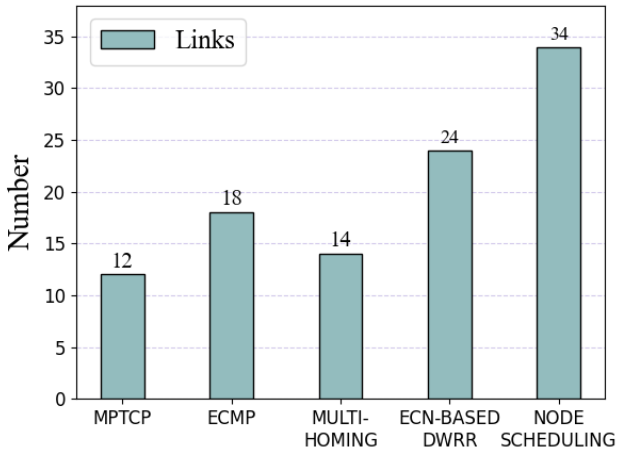


Figure 10. Number of transmission links.

6.3.2. Average Bandwidth Utilization

Figure 11 shows the average bandwidth utilization of the network links under five multipath scheduling strategies. The MPTCP scheme uses the fewest transmission links, and the default scheduling policy evenly distributes traffic to each path, resulting in the lowest average bandwidth utilization. ECMP distributes traffic evenly across all equivalent paths, and as the number of paths used increases, the bandwidth utilization is improved compared to MPTCP. The MULTI-HOMING scheme has a similar number of links as the MPTCP scheme. Still, at each network node, the port with the lowest load can be selected for forwarding based on the status of multiple IPs corresponding to multiple ports, resulting in a 45% increase in bandwidth utilization compared to MPTCP. ECN-BASED DWRR is not limited to equivalent links, and its bandwidth utilization is about twice that of the MPTCP scheme. The scheme proposed in this article uses the highest number of links for transmission, and each node can be scheduled based on link status and feedback information. The bandwidth utilization rate is 69% higher than ECN-BASED DWRR and about 3.4 times that of the MPTCP scheme.

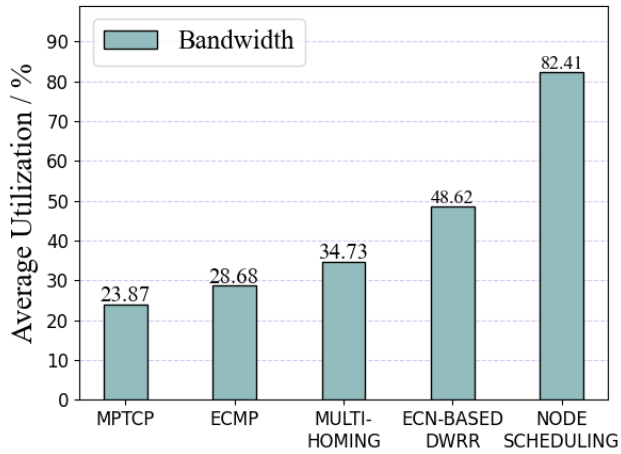


Figure 11. Average bandwidth utilization.

6.4. Network Throughput

In order to visually demonstrate the throughput performance of five multipath scheduling schemes, we conducted comparative experiments on overall throughput and user throughput. The experimental results are reported as follows.

6.4.1. Overall Throughput

We instructed the server to send data to four user nodes simultaneously and changed the network bandwidth by controlling the proportion of available bandwidth in the network. We calculated the changes in network throughput among the five schemes, and Figure 12 shows the experimental results.

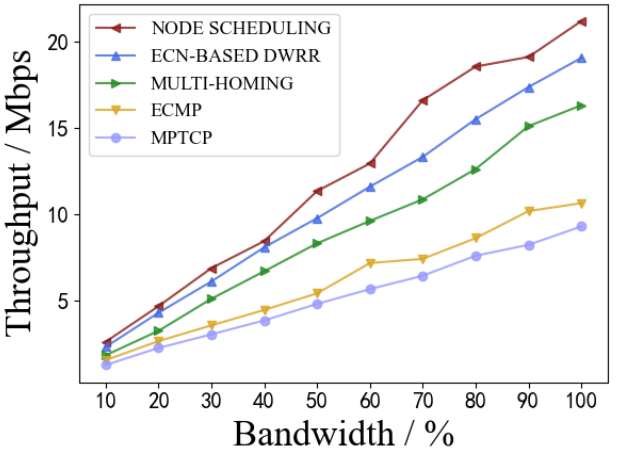


Figure 12. Overall average throughput.

Figure 12 shows the variation of average network throughput with network bandwidth. The throughput of all five multipath scheduling schemes increases with the increase in network bandwidth and the enhancement of network transmission capacity. The throughput gap between the five schemes gradually increases as the bandwidth increases. The MPTCP scheme has the lowest throughput. When the bandwidth reaches 100%, the throughput of ECMP increases by 14% compared to MPTCP, the throughput of MULTI-HOMING increases by 75% compared to MPTCP, and the throughput of ECN-BASED DWRR is about twice that of MPTCP. The throughput of the proposed scheme is about 2.2 times that of MPTCP, which is 11% higher than ECN-BASED DWRR.

6.4.2. User Throughput

We instructed the Server to send data to four user nodes and changed the network bandwidth by controlling the proportion of available bandwidth in the network. The throughput changes of four users were recorded, and Figure 13 shows the experimental results.

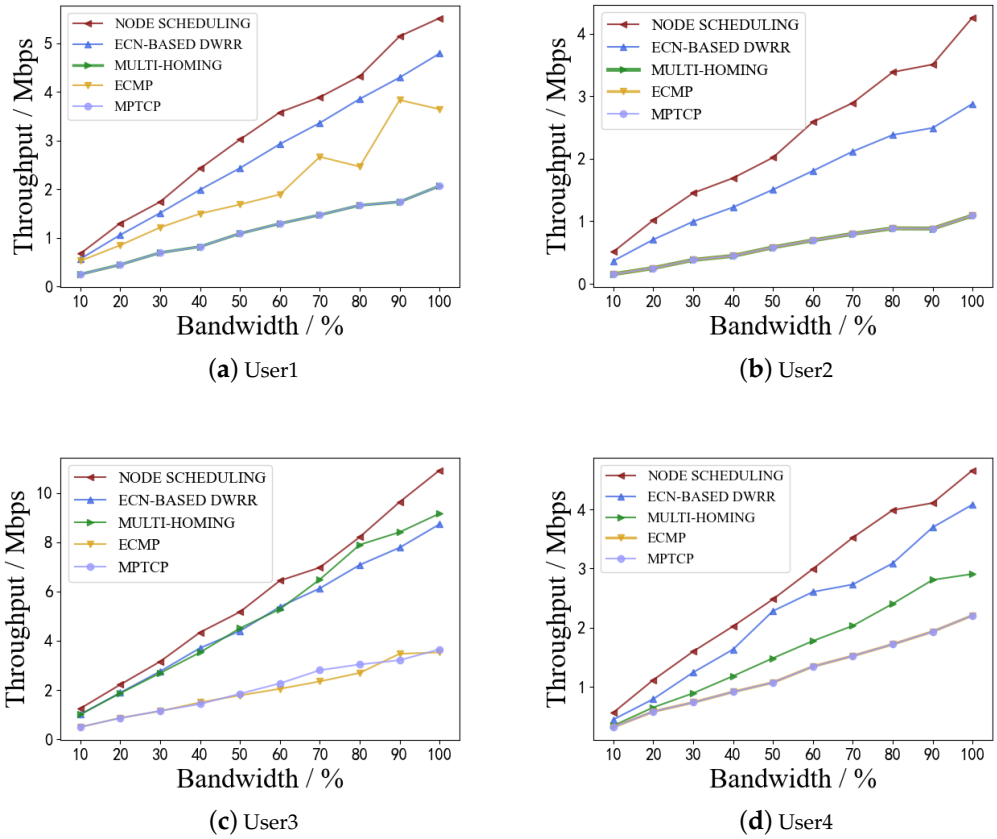


Figure 13. User average throughput.

Figure 13 shows four users' throughput as a network bandwidth function. The throughput of all five multipath scheduling schemes increases with the increase in network bandwidth and the enhancement of network transmission capacity. By comparing the throughput performances between User3 and User4 and between User1 and User2, it can be seen that when a user only has one IP, the throughput of MULTI-HOMING is similar to that of MPTCP, and they become the shortest path transmissions. By comparing the throughput performance of User1 and User2, it can be seen that the ECMP scheme's throughput performance depends on the network's equivalent path. The ECMP scheme also becomes the shortest path transmission when there is no equivalent path. ECN-BASED DWRR and the proposed scheme are not limited by the number of receiver IPs and equivalent paths, and their throughput performance is improved compared to the MPTCP scheme in all users. Moreover, the throughput performance of the proposed scheme is better than that of ECN-BASED DWRR.

6.5. Average Flow Completion Time

To visually display the average flow completion time of five multipath scheduling schemes, we instructed the server node to send 100 MB of data to four users simultaneously. We calculated the average flow completion time (FCT) of four users in different schemes. The experimental results are as reported as follows.

Figure 14 shows the variation in average flow completion time with link bandwidth. As network bandwidth increases and network transmission capacity is enhanced, the aver-

age flow completion time of the five multipath scheduling schemes decreases. The average flow completion time of the MPTCP scheme is the longest. When the bandwidth reaches 100%, the average transmission time of ECMP is 87% of MPTCP, the average transmission time of MULTI-HOMING is 57% of MPTCP, the average transmission time of ECN-BASED DWRR is 48% of MPTCP, and the average transmission time of the proposed scheme is 43% of MPTCP, which is 10% less than ECN-BASED DWRR.

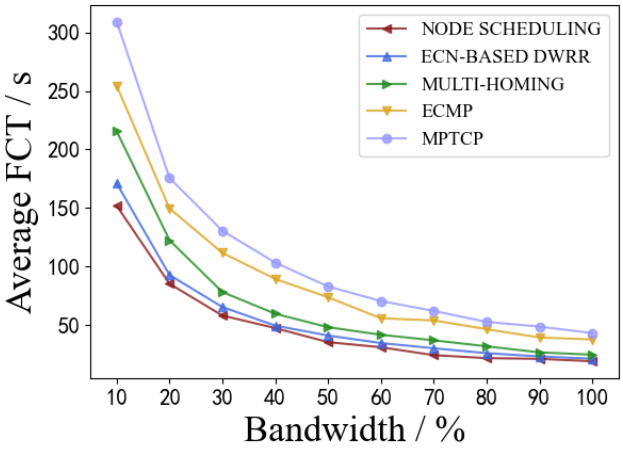


Figure 14. Average flow completion time.

7. Discussion

7.1. Control Cycle

This article only qualitatively evaluates the control cycle, and the longer the port queue, the longer the control cycle can be. The optimal control cycle with the best scheduling effect can be obtained by conducting simulation experiments. In actual networks, obtaining the optimal control cycle can be more complex when the network size is large.

7.2. Packet Disorder

At present, the path scheduling scheme on network nodes proposed in this article can better utilize the diversity of network paths. Using more paths for transmission results in greater differences between paths, which can easily cause packet disorder. Although there are many studies on disordered recombination at the receiving end, an increase in the degree of disorder increases the recombination time.

7.3. Future Research

In future research, we will further optimize the algorithm to calculate control cycles, improve the accuracy of control cycle calculations, and achieve better scheduling performance. At the same time, we will consider adjusting the scheduling granularity in the process of path scheduling, transferring a group of consecutive packets to the same link, reducing the degree of packet disorder, and reducing the time consumed by the reordering of the receiver.

8. Conclusions

This study introduces a novel strategy, i.e., multipath scheduling, for management of network traffic. This method involves creating a set of potential next-hop nodes chosen through a combination of global and localized routing strategies. Traffic is then distributed among these nodes in varying proportions for efficient management. The status of the network links is regularly updated, which, in turn, allows for the adjustment of traffic allocations to each node in the set. If scheduling issues arise within this set, feedback is relayed to the nodes upstream. Moreover, this paper presents a specific design of incremental deployment using overlay. The effectiveness of this approach was verified through

simulation experiments. The results indicate that the link-status-based scheduling method surpasses traditional methods such as random, polling, and free-bandwidth-based methods in terms of throughput efficiency. When compared with other solutions, such as MPTCP, ECMP, MULTI-HOMING, and ECN-BASED DWRR, the proposed method exhibits superior network link utilization. This translates to enhanced overall network bandwidth usage, achieving an impressive 82.41% utilization rate. Regarding throughput, under similar network bandwidth scenarios, the proposed method significantly outperforms MPTCP, offering more than double its throughput. Moreover, this strategy is not limited by the number of user addresses or available paths, making it a more flexible solution. In terms of transmission time, the method achieves the quickest results under equivalent network resource conditions, cutting down transmission time by 57% compared to MPTCP.

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

Funding: This work was funded by the Strategic Priority Research Program of the Chinese Academy of Sciences: Information Collaborative Service and Data Sharing (Grant No. XDA031050100).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: We would like to express our gratitude to Hong Ni, Rui Han, Yong Xu, and Younan Lu for their meaningful support for this work.

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

References

1. Zhang, C. Research on the Optimization Strategy of Traffic Capacity Based on BA Scale-Free Networks. Master's Thesis, Xidian University, Xi'an, China, 2018.
2. Ling, X.; Jiang, R.; Wang, X.; Hu, M.; Wu, Q. Traffic of packets with non-homogeneously selected destinations in scale-free network. *Phys. A Stat. Mech. Appl.* **2008**, *387*, 4709–4715. [CrossRef]
3. Yang, H.; Wang, W.; Wu, Z.; Wang, B. Traffic dynamics in scale-free networks with limited packet-delivering capacity. *Phys. A Stat. Mech. Appl.* **2008**, *387*, 6857–6862. [CrossRef]
4. Liu, Z.; Hu, M.B.; Jiang, R.; Wang, W.X.; Wu, Q.S. Method to enhance traffic capacity for scale-free networks. *Phys. Rev. E* **2007**, *76*, 037101. [CrossRef]
5. Wang, W.X.; Yin, C.Y.; Yan, G.; Wang, B.H. Integrating local static and dynamic information for routing traffic. *Phys. Rev. E* **2006**, *74*, 016101. [CrossRef]
6. Ahlgren, B.; D'Ambrosio, M.; Marchisio, M.; Marsh, I.; Dannewitz, C.; Ohlman, B.; Pentikousis, K.; Strandberg, O.; Rembarz, R.; Vercellone, V. Design considerations for a network of information. In Proceedings of the the 2008 ACM CoNEXT Conference, Madrid, Spain, 9–12 December 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 1–6.
7. 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, Rome, Italy, 1–4 December 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 1–12.
8. Zhang, L.; Afanasyev, A.; Burke, J.; Jacobson, V.; Claffy, K.; Crowley, P.; Papadopoulos, C.; Wang, L.; Zhang, B. Named Data Networking. A*CM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 66–73. [CrossRef]
9. Wang, J.; Cheng, G.; You, J.; Sun, P. SEANet: Architecture and Technologies of an On-site, Elastic, Autonomous Network. *J. Netw. New Media* **2020**, *6*, 1–8.
10. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76. [CrossRef]
11. Ford, A.; Raiciu, C.H.; Ley, M.; Bonaventure, O. TCP Extensions for Multipath Operation with Multiple Addresses. 2013. Available online: https://www.rfc-editor.org/rfc/rfc6824.html (accessed on 29 July 2023).
12. Stewart, R. Stream Control Transmission Protocol. 2007. Available online: https://www.rfc-editor.org/rfc/rfc4960 (accessed on 29 July 2023).
13. Zheng, Z.; Ma, Y.; Liu, Y.; Yang, F.; Li, Z.; Zhang, Y.; Zhang, J.; Shi, W.; Chen, W.; Li, D.; et al. Xlink: Qoe-driven multi-path quic transport in large-scale video services. In Proceedings of the 2021 ACM SIGCOMM 2021 Conference, Virtual Event, 23–27 August 2021; pp. 418–432.
14. De Coninck, Q.; Bonaventure, O. Multipath quic: Design and evaluation. In Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies, Incheon, Republic of Korea, 12–15 December 2017; pp. 160–166.

15. Özcan, Y.; Guillemin, F.; Houzé, P.; Rosenberg, C. Fast and smooth data delivery using MPTCP by avoiding redundant retransmissions. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7.
16. Hwang, J.; Yoo, J. Packet scheduling for multipath TCP. In Proceedings of the 2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo, Japan, 7–10 July 2015; pp. 177–179.
17. Le, T.A.; Bui, L.X. Forward delay-based packet scheduling algorithm for multipath TCP. *Mob. Netw. Appl.* **2018**, *23*, 4–12. [CrossRef]
18. Chung, J.; Han, D.; Kim, J.; Kim, C. Machine learning based path management for mobile devices over MPTCP. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Republic of Korea, 13–16 February 2017; pp. 206–209.
19. Sunshine, C.A. Source routing in computer networks. *ACM SIGCOMM Comput. Commun. Rev.* **1977**, *7*, 29–33. [CrossRef]
20. Hu, Y.-C.; Johnson, D.B. Implicit source routes for on-demand ad hoc network routing. In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Long Beach, CA, USA, 4–5 October 2001; pp. 1–10.
21. Bellovin, S.M. Security problems in the TCP/IP protocol suite. *ACM SIGCOMM Comput. Commun. Rev.* **1989**, *19*, 32–48. [CrossRef]
22. Pióro, M.; Szentesi, Á; Harmatos, J.; Jüttner, A.; Gajowniczek, P.; Kozdrowski, S. On Open Shortest Path First Related Network Optimisation Problems. *Performance Eval.* **2002**, *48*, 201–223. [CrossRef]
23. Fortz, B.; Umit, H. Efficient Techniques and Tools for Intra-domain Traffic Engineering. *Int. Trans. Oper. Res.* **2011**, *18*, 359–376. [CrossRef]
24. Harmatos, J. A Heuristic Algorithm for Solving the Static Weight Optimisation Problem in OSPF Networks. In Proceedings of the GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270), San Antonio, TX, USA, 25–29 November 2001; pp. 1605–1609.
25. Sridharan, A.; Gu´erin, R.; Diot, C. Achieving Near-optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks. *IEEE/ACM Trans. Netw.* **2005**, *13*, 234–247. [CrossRef]
26. Zhang, J.; Xi, K.; Zhang, L.; Chao, H.J. Optimizing Network Performance Using Weighted Multipath Routing. In Proceedings of the 2012 21st International Conference on Computer Communications and Networks (ICCCN), Munich, Germany, 30 July–2 August 2012; pp. 1–7.
27. Yan, G.; Zhou, T.; Hu, B.; Fu, Z.Q.; Wang, B.H. Efficient routing on complex networks. *Phys. Rev. E* **2006**, *73*, 046108. [CrossRef] [PubMed]
28. Dong, J.Q.; Huang, Z.G.; Zhou, Z.; Huang, L.; Wu, Z.X.; Do, Y.; Wang, Y.H. Enhancing transport efficiency by hybrid routing strategy. *Europhys. Lett.* **2012**, *99*, 20007. [CrossRef]
29. Pu, C.L.; Zhou, S.Y.; Wang, K.; Zhang, Y.F.; Pei, W.J. Efficient and robust routing on scale-free networks. *Phys. A Stat. Mech. Appl.* **2012**, *391*, 866–871. [CrossRef]
30. Liu, Q.; Zhang, D. Routing strategy research based on mixed information on complex networks. *Comput. Eng. Des.* **2012**, *33*, 880–884.
31. Echenique, P.; Gómez-Gardeñes, J.; Moreno, Y. Improved routing strategies for Internet traffic delivery. *Phys. Rev. E* **2004**, *70*, 056105. [CrossRef]
32. Echenique, P.; Gómez-Gardenes, J.; Moreno, Y. Dynamics of jamming transitions in complex networks. *Europhys. Lett.* **2005**, *71*, 325. [CrossRef]
33. Tang, M.; Liu, Z.; Liang, X.; Hui, P.M. Self-adjusting routing schemes for time-varying traffic in scale-free networks. *Phys. Rev. E* **2009**, *80*, 026114. [CrossRef]
34. Yu, M. Network telemetry: Towards a top-down approach. *ACM SIGCOMM Comput. Commun. Rev.* **2019**, *49*, 11–17. [CrossRef]
35. Morton, A. Active and Passive Metrics and Methods (with Hybrid Types In-Between). Fremont: IETF. 2016. Available online: https://www.rfc-editor.org/rfc/rfc7799.html (accessed on 29 July 2023).
36. Riesenberg, A.; Kirzon, Y.; Bunin, M.; Galili, E.; Navon, G.; Mizrahi, T. Time-multiplexed parsing in marking-based network telemetry. In Proceedings of the 12th ACM International Conference on Systems and Storage, Haifa, Israel, 3–5 June 2019; pp. 80–85.
37. Karaagac, A.; De Poorter, E.; Hoebeke, J. Alternate marking-based network telemetry for industrial WSNs. In Proceedings of the 2020 16th IEEE International Conference on Factory Communication Systems (WFCS), Porto, Portugal, 27–29 April 2020; pp. 1–8.
38. Kagami, N.S.; da Costa Filho, R.I.T.; Gaspary, L.P. Capest: Offloading network capacity and available bandwidth estimation to programmable data planes. *IEEE Trans. Netw. Serv. Manag.* **2019**, *17*, 175–189. [CrossRef]
39. Yousaf, F.Z.; Bredel, M.; Schaller, S.; Schneider, F. NFV and SDN—Key technology enablers for 5G networks. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2468–2478. [CrossRef]
40. Ordonez-Lucena, J.; Ameigeiras, P.; Lopez, D.; Ramos-Munoz, J.J.; Lorca, J.; Folgueira, J. Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. *IEEE Commun. Mag.* **2017**, *55*, 80–87. [CrossRef]
41. Bifulco, R.; Rétvári, G. A survey on the programmable data plane: Abstractions, architectures, and open problems. In Proceedings of the 2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR), Bucharest, Romania, 18–20 June 2018; pp. 1–7.
42. Han, S.; Jang, S.; Choi, H.; Lee, H.; Pack, S. Virtualization in programmable data plane: A survey and open challenges. *IEEE Open J. Commun. Soc.* **2020**, *1*, 527–534. [CrossRef]

43. Bosshart, P.; Daly, D.; Gibb, G.; Izzard, M.; McKeown, N.; Rexford, J.; Schlesinger, C.; Talayco, D.; Vahdat, A.; Varghese, G.; et al. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 87–95. [CrossRef]
44. Li, S.; Hu, D.; Fang, W.; Ma, S.; Chen, C.; Huang, H.; Zhu, Z. Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability. *IEEE Netw.* **2017**, *31*, 58–66. [CrossRef]
45. Li, X.; Wu, Y.; Ge, J.; Zheng, H.; Yuepeng, E.; Han, C.; Lv, H. A kernel-space POF virtual switch. *Comput. Electr. Eng.* **2017**, *61*, 339–350. [CrossRef]
46. Berde, P.; Gerola, M.; Hart, J.; Higuchi, Y.; Kobayashi, M.; Koide, T.; Lantz, B.; O'Connor, B.; Radoslavov, P.; Snow, W. ONOS: Towards an open, distributed SDN OS. In Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, Chicago, IL, USA, 22 August 2014; pp. 1–6.
47. Ma, P.; You, J.; Wang, J. An efficient multipath routing schema in multi-homing scenario based on protocol-oblivious forwarding. *Front. Comput. Sci.* **2020**, *14*, 144501. [CrossRef]
48. Motohashi, H.; Le Nguyen, P.; Nguyen, K.; Sekiya, H. Implementation of P4-Based Schedulers for Multipath Communication. *IEEE Access* **2022**, *10*, 76537–76546. [CrossRef]
49. Nadeem, K.; Jadoon, T.M. An ns-3 MPTCP Implementation. In *Quality, Reliability, Security and Robustness in Heterogeneous Systems, Proceedings of the 14th EAI International Conference, Qshine 2018, Ho Chi Minh City, Vietnam, 3–4 December 2018*; Proceedings 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 48–60.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.