



Article A Reverse Shortest Path Tree-Based Multicast Joining Node Selection Method

Zhenyu Tian ^{1,2}, Jiali You ^{1,2,3} and Linlin Hu ^{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; tianzy@dsp.ac.cn
- ² School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, China
- ³ Peng Cheng Laboratory, Xili Road, Nanshan District, Shenzhen 518055, China
- * Correspondence: hull@dsp.ac.cn

Abstract: Network layer multicast is a powerful method for transmitting data from sources to multiple group members. When joining a multicast group, a group member first sends a request to a designated router (DR). Then, the DR selects a node in the existing multicast tree (known as a multicast joining node, or MJN) to establish a multicast distribution path from the MJN to itself. The MJN selection method runs on the DR and has a significant impact on the distribution of the multicast tree, that directly affects the load distribution in the network. However, the current MJN selection method cannot effectively detect the load status of the downlink multicast path in the case of asymmetric routing, leading to network congestion and limiting the number of multicast groups that the network can accommodate (multicast capacity). To solve this problem, we propose an MJN selection method based on the reverse shortest path tree (RSPT). RSPT can effectively detect the load status of downlink multicast paths in case of routing asymmetry. Based on the detection results of RSPT, DR can select the MJN with the lowest path load to join the multicast tree. Our experimental results indicate that compared to existing multicast methods, our method has a lower cost and delay, and can effectively balance the network load in the case of asymmetric routing, increasing multicast capacity by more than two times.

Keywords: network layer multicast; multicast joining node; asymmetric routing;multicast capacity

1. Introduction

With the continuous advancement of multimedia technology, the volume of traffic generated by multimedia applications, such as online meetings and live video broadcasts, has been increasing steadily [1–3]. Multicast is a data transmission method that transmits packets along a distribution tree from data sources to a set of receivers by simply replicating the packets at branches of the multicast distribution tree [4–6]. Thus, multicast can effectively save network resources and improve bandwidth utilization. Multicast technology can be divided into network layer multicast, application layer multicast, and link layer multicast. Compared with the other two technologies, network layer multicast has greater scalability, as it has the ability to effectively avoid redundant data transmission and cover a wider network range. As a result, network-layer multicast has always been an important research topic.

According to Cisco's annual internet report [7], the number of devices connected to the internet worldwide is expected to increase by 50% from 2018 to 2023. With the growth in the number of network devices, the number of multicast groups on the network is also expected to increase. In multicast technology, the multicast capacity, which refers to the number of multicast groups that the network can accommodate, is a very important indicator. The multicast capacity signifies the throughput and efficiency of the network, and is an important measure of the scalability of multicast technology.



Citation: Tian, Z.; You, J.; Hu, L. A Reverse Shortest Path Tree-Based Multicast Joining Node Selection Method. *Future Internet* **2023**, *15*, 156. https://doi.org/10.3390/fi15050156

Academic Editor: Franco Davoli

Received: 30 March 2023 Revised: 14 April 2023 Accepted: 18 April 2023 Published: 23 April 2023



Copyright: © 2023 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/). of multicast joining nodes (MJNs) has a significant influence on multicast capacity. When joining a multicast group, a group member first sends a request to a designated router (DR). Then, the DR selects a node in the existing multicast tree (known as a multicast joining node, or MJN) to establish a multicast distribution path from the MJN to itself. The MJN selection method running on the DR has a significant impact on the distribution of the multicast tree, which directly affects the load distribution in the network, and thus has a considerable impact on multicast capacity [8–10].

The main challenge faced by MJN selection methods is how to effectively detect the path load status between MJN and DR, especially in the case of asymmetric routing [4,8,9]. Due to the fact that asymmetric routing has become a common feature of networks, more and more multicast technologies are adopting a downlink multicast path to ensure the quality of multicast data transmission [1,11–13]. The establishment of downlink paths means that the multicast path is the shortest path from MJN to DR, which ensures that data can be transmitted from the multicast tree to the receiver with the shortest delay. In contrast to the downlink path, the uplink path means that the multicast path is the shortest path from DR to MJN, and multicast data will not be transmitted from the multicast tree to the receiver with the shortest delay, leading to a deterioration in the quality of multicast service. In order to establish low-load downlink multicast paths, DR must consider asymmetric routing when detecting multicast path load status, otherwise the load status of the upstream path will be detected, resulting in incorrect detection results. However, few existing MJN selection methods consider this. This defect makes it difficult for existing MJN selection methods to achieve good results in real environments. In addition, MJN selection methods should have low latency and distributed characteristics for practical engineering deployment.

To address this issue, we have designed an MJN selection method based on the reverse shortest path tree (RSPT). During network initialization, the DR constructs an RSPT of K-hop subgraph with itself as the root, that is composed of all shortest paths from other nodes in the subgraph to the DR. As long as the leaf nodes of RSPT send probe messages to the DR, the DR can perceive the load status of all downlink paths and select the MJN with the lowest load to join the multicast tree. Therefore, RSPT can help the DR efficiently establish downlink multicast paths in the case of asymmetric routing. In addition, the RSPT-based MJN selection method is distributed, low-latency, and has the potential for practical engineering deployment.

The main contributions of this paper are:

- This article proposes an efficient load detection mechanism for downlink multicast paths. By constructing RSPT, this mechanism enables DR to evaluate the load status of downlink multicast paths efficiently. This mechanism is fully applicable to situations where routing is asymmetric;
- 2. We have designed a real-time, distributed MJN selection algorithm that can immediately provide service for new group members, ensuring minimal delay in multicast service. Additionally, this algorithm does not require a central management node to participate in the calculation of multicast paths, which helps improve the scalability of multicast;
- 3. Through a large number of experiments, the proposed method has been shown to have a low cost, low latency, and the ability to effectively balance the load state in the network, even the routing is asymmetric.

The structure of this article is organized as follows. Section 2 presents a review of the related work. Section 2 introduces the proposed MJN selection method, which consists of two main parts: the construction of RSPT and the implementation of the MJN selection method. Section 4 describes the experiments conducted to evaluate the proposed method. Finally, Section 5 concludes the paper. The notations and definitions used in this paper are summarized in Table 1.

Notation	Definition		
MJN	Multicast Joining Node		
RP	Rendezvous Point		
SBT	Source Based Tree		
ST	Shared Tree		
MFT	Multicast Forwarding Table		
DR	Designated Router		
NRS	Name Resolution System		
NA	Network Address		
MID	Multicast Group ID		

Table 1. Notations and definitions.

2. Related Work

In this chapter, we focus on traditional IP multicast and existing ICN multicast. In traditional TCP/IP multicast, WAVE [13] is an important MJN selection method that influences many multicast mechanisms [14–17].

This method requires the DR to first send a request to the root of the multicast tree. When the multicast tree root receives the request, it will forward the request along the multicast tree. Each multicast tree node that receives the request will reply to the DR with a probe message, that will collect the load status information along the way. Upon receiving the probe messages, the DR can understand the load status of the downlink multicast paths, and then select the path with the lowest load to join the multicast tree. This method is essentially a greedy method that selects the path with the lowest load to join the multicast tree each time. It can effectively deal with situations where the routing is asymmetric. However, because it requires probing all of the nodes on the multicast tree, this method can lead to long multicast service delays, heavy network loads, and difficulty in deploying in engineering environments.

In order to meet the deployment requirements in engineering environments, many mature TCP/IP multicast techniques directly select the root node of the multicast tree as the MJN. Among them, some multicast technologies construct source trees, in which the root node of the multicast tree is located at the data source. Therefore, the MJNs of these multicast technologies are data sources, such as PIM-SSM [18], PIM-DM [19], MOSPF [20], etc. There are also some multicast technologies that construct shared trees, in which the root node of the multicast tree is located at the core or Rendezvous Point(RP). Therefore, the MJNs of these multicast technologies are all cores or RPs. These MJN selection methods cannot consider the network load status, which may easily lead to network load congestion and limit the scalability of multicast.

In ICN, routing is based on the content name rather than the IP address. This namebased routing allows receivers to request data from any position on the multicast tree, which is beneficial for optimizing the path of multicast data transmission [21–24]. The following is an introduction to some of the MJN selection methods used in ICN multicast technologies.

ILDM [25] is a multicast technology based on ICN. This technology proposes a pathstate-aware MJN selection method. The method requires the control plane to calculate the path from MJN to DR and to detect various load statuses. Upon obtaining the load status of the downlink multicast path, the control plane returns the MJN with the lowest path load to DR. The MJN selection method of ILDM is essentially the same as WAVE, both of which adopt a greedy strategy to select the path with the lowest load to join the multicast tree. The difference is that the path detection of WAVE is distributed, while ILDM utilizes the control plane. ILDM can effectively deal with asymmetric routing issues. However, ILDM is a centralized MJN selection mechanism, that requires the control plane to perform a large amount of calculation, resulting in long multicast service delays and high pressure on the central node.

Classic ICN solutions, such as DONA [26], CCN [27], and NDN [28,29] inherently support multicast. The multicast mechanism of these solutions aggregates identical data

requests by preserving existing data forwarding paths, thus supporting multicast services. When selecting MJN in these multicast solutions, they essentially choose the node closest to the receiver. Data request packets sent by the receiver will be forwarded in the network according to specific rules, and the first node that has the requested data will become the MJN. This method is advantageous in shortening data forwarding paths, but it cannot perceive the network's load state, which can lead to congestion.

As a famous ICN solution, PURSUIT [30] also proposes a multicast mechanism. The multicast mechanism in PURSUIT relies on the topology manager (TM). TM in PURSUIT encodes the forwarding path in the Bloom filter of the data packet. When the data packet reaches a forwarding node, the forwarding node only needs to perform an "AND" operation between the label of the outgoing link and the Bloom filter in the data packet. If there is any match, the data packet is forwarded through the corresponding link. Multicast transmission can be achieved by simply encoding the entire multicast tree into a single Bloom filter. This multicast method relies on the central management node TM. However, this method does not take into account the issue of routing asymmetry when calculating multicast trees. Moreover, the centralized calculation method is not conducive to improving multicast scalability.

NOMA [31] is a multicast communication mechanism designed based on MobilityFirst. NOMA uses Global Name Resolution Service (GNMRS) as a mapping between identifiers and address locators. NOMA first assigns a unique name to each network node; all entities interested in receiving data from a multicast stream can register their unique name in GNMRS. Then, the multicast service manager near the gateway uses this information to construct a multicast tree based on available resources and the desired size of the tree. Then, recursively, the names of each branch router in the multicast tree are mapped to the multicast group identifier in GNMRS. Similar to ILDM, PURSUIT, etc., the construction and maintenance of the multicast tree in NOMA depends on the multicast management node. However, this method does not take into account the issue of routing asymmetry when calculating multicast trees. Moreover, the centralized calculation method is not conducive to improving multicast scalability. At the same time, when the multicast membership changes, this method requires recalculation of the multicast tree, which can have a significant impact on the quality of multicast services.

In addition, there are many methods attempting to construct Steiner trees to the balance network load and improve multicast capacity. These methods include algorithms based on genetic algorithms [14,15,32], taboo search [33], greedy randomized adaptive search process [16,34], simulated annealing [17,35], and other ideas. Many of these algorithms also use greedy strategies proposed in WAVE [14–17]. However, a common problem with these algorithms is that they rely on centralized computing paradigms and involve a large number of iterative calculations. This can lead to significant service delays and central node performance bottlenecks, which hinder the scalability of multicast. In addition, they have not been able to address the issue of routing asymmetry.

In summary, some of the existing MJN selection methods are unable to consider the asymmetry of routing, such as most TCP/IP multicast protocols; some methods have high multicast processing delays and heavy network loads, such as WAVE; and some methods use a centralized computing paradigm, that is not conducive to improving multicast scalability, such as ILDM, PURSUIT, and NOMA. Therefore, there is a need for an MJN selection method that fully considers the asymmetry of routing, has low latency, and distributed characteristics.

3. Reverse Shortest Path Tree Based MJN Selection Method

Our RSPT-based MJN selection method can be divided into two parts. The first part involves constructing the RSPT, while the second part involves utilizing network load state information collected by RSPT to select the MJN.

3.1. Construction of RSPT

RSPT is composed of the shortest paths from other nodes in the K-hop subgraph to DR. DR can efficiently evaluate the load status of the downlink multicast path using RSPT, even in cases of routing asymmetry. DR can further select the MJN that minimizes the cost of the multicast path to join the multicast tree, reducing network load congestion and improving multicast scalability. Therefore, RSPT is a key component of our proposed method.

The construction of RSPT consists of three main steps: firstly, the construction of a K-hop subgraph, secondly, the collection of information using probe messages, and lastly, the generation of RSPT.

3.1.1. The K-Hop Sub-Graph Construction

DR utilizes RSPT to the detect network load, but it may become overwhelmed when dealing with larger networks. To mitigate this, DR will only construct an RSPT that covers its own K-hop neighbourhood subgraph. During the network initialization phase, DR will broadcast requests with a maximum forwarding distance of K. This ensures that all K-hop neighbours of DR receive the requests. According to the "Six Degrees of Separation" principle, it is possible to connect with anyone within six hops in a social network. Therefore, we limit the value of K to be less than six.

3.1.2. Using Probe Message to Collect Information

Upon receiving a probe request, each node responds with a probe message that is sent to the DR via unicast. The nodes forwarding the probe message add their own load status to it. In large network topologies, the probe messages may contain excessive node information that exceeds the maximum transmission unit (MTU) limit. Therefore, the probe message supports fragmentation. The design of the probe message is illustrated in Figure 1.

Type Fragment Fragment Noo Tag Index Cou	Node Index I
---	---

Figure 1. Probe message design.

The header of the probe message has the following fields.

- 1. Type: This field indicates the type of the message, whether it is a probe message or a request message;
- 2. Fragment Tag: This flag indicates whether the message is the last fragment or not. If the message does not have any fragments or is the last one, the flag is set to 1; otherwise, it is set to 0;
- 3. Fragment Index: This parameter indicates the index of the current fragment;
- 4. Node Cnt: This parameter indicates the maximum number of node information that can be stored in the message;
- 5. Node index: This parameter indicates the number of nodes with information that has been recorded in the message;
- 6. (IM ID-load status): Each intermediate node's information is stored as a key-value pair, with the ID of the node as the key and its load status as the value.

$$\operatorname{Cost}(i,j) = \begin{cases} \alpha \frac{\operatorname{dis}(i,j)}{\operatorname{Diameter}} + \beta \frac{\operatorname{PT}}{T_{\max}} + \gamma \frac{\operatorname{PL}}{L_{\max}} + \delta \frac{\operatorname{avlT}}{T_{\max}} + \epsilon \frac{\operatorname{avlL}}{L_{\max}} & \operatorname{dis}(i,j) \le K \\ \zeta \frac{\operatorname{dis}(i,j)}{\operatorname{Diameter}} & \operatorname{else} \end{cases}$$
(1)

$$\begin{aligned} \alpha + \beta + \gamma + \delta + \epsilon &= 1 \\ 0 \leq \alpha, \beta, \gamma, \delta, \epsilon \leq 1 \\ \zeta \geq 1 \end{aligned}$$
 (2)

3.1.3. The Generation of RSPT

To reduce the number of probe messages, we construct a reverse shortest path tree(RSPT) with the DR as the root. Once DR establishes RSPT, DR can know all of the leaf nodes. Then, DR will periodically send probe requests to the leaf nodes. When nodes receive the probe request, they will know that they are the leaf nodes of the RSPT, and then reply to the DR with probe messages.

When the DR receives a probe message, it can extract a node sequence *Seq* from it. The DR then checks whether *Seq* is included in the existing RSPT *C*. If *Seq* is included in *C*, it is discarded. Otherwise, *Seq* is added to *C*, and any sequence in *C* that is contained by *Seq* is removed. *C* is a set of sequences, where *C*, including *Seq*, means that *Seq* is a subsequence of any sequence in *C*.

Once the DR receives the probe messages of all nodes, the first node of all sequences in *C* are leaf nodes in the reverse path tree. The algorithm is shown in Algorithm 1.

Algorithm 1 The RSPT Generation Algorithm.

```
C \leftarrow \emptyset
for each new vector S do
    InsertLeaf(S, C)
end for
return C
procedure INSERTLEAF(S,C)
    flag \leftarrow 0
    for each T in C do
       if S is in T then
            flag \leftarrow 1
           break
       else if T is in S then
           delete T from C
           if flag == 0 then
               insert S in C
               flag \leftarrow 1
           end if
       end if
    end for
    if flag == 0 then
       insert S in C
    end if
end procedure
```

3.1.4. An Example of RSPT

In Figure 2, node *a* constructs the RSPT with itself as the root node. It first floods requests to all other nodes in the network. Each node that receives the request will send the probe messages back to node *a*. By following Algorithm 1, the RSPT can be constructed with nodes *c*, *e*, and *g* as the leaves.



Figure 2. RSPT example.

3.2. MJN Selection Method Implementation

In this section, we will discuss the process of selecting the most suitable MJN based on the information collected from the RSPT. By using the information gathered from RSPT, DR can calculate the downlink path cost related to each MJN, even when routing is asymmetric. The equation for calculating the path cost is shown in Equation (1). DR will then choose the MJN with the lowest path cost to join the multicast tree.

In Equation (1), we define several parameters to calculate the cost of a multicast path. The distance between nodes *i* and *j* is represented by dis(i, j), while the topological diameter of the network is denoted by *Diameter*. The reason for designing this parameter is that longer paths put more pressure on the network, so they should have a higher cost.

PT, *PL*, *aveT*, and *aveL* are four parameters that represent the load on the path. *PT* represents the number of MFT entries on the node with the highest load on the path. *PL* represents the bandwidth usage of the link with the highest load on the path. *aveT* means the average number of MFT entries recorded by all nodes on the path. *aveL* represents the average bandwidth utilization of all links on the path. The reason for considering these load parameters is to reduce load congestion and prevent paths from passing through nodes and links with high loads.

To normalize various types of loads, we introduce a maximum forwarding state quantity T_{max} that a node can handle and a maximum load L_{max} that a link can handle.

Next, we will introduce the determination method for each weight. In Formula (1), the values of β and γ are both 0.3, the value of α is 0.2, and the values of σ and ϵ are both 0.1. As PT and PL are the parameters that most intuitively measure the load status of the highest-loaded points and edges on the path, setting β and γ to be larger can help avoid congestion of individual nodes and links, delaying the time to reach the network performance bottleneck. α is also important. Setting it to be larger can help shorten the length of the multicast path and reduce the consumption of network resources. σ and ϵ measure the average load status of the path, which also helps to delay the time for the network to reach the performance bottleneck. However, through experiments, it has been found that their importance is not as significant as other parameters, so the values of σ and ϵ are relatively small.

In Formula (2), as the topology size increases, ζ also increases accordingly. This is because in large topologies, in order to avoid excessive load concentration on nodes outside the detection range, each DR prioritizes selecting nodes within its own detection range. Since DR cannot obtain the path load status of nodes outside the detection range, it can easily lead to load congestion problems.

Upon computing the path cost for each MJN, DR selects the node with the minimum cost to join the multicast tree.

It can be seen that DR can make real-time decisions upon receiving join requests by utilizing cached information, thus ensuring low multicast service latency. Additionally, the method proposed does not require the participation of a central management node, making it a purely distributed approach that avoids central bottleneck issues and enhances multicast scalability.

4. Simulation

We used the Waxman model to generate four topological structures, each containing 200 nodes (Waxman200), 2k nodes (Waxman2000), 200k nodes (Waxman200k), and 2 million nodes (Waxman2000k). When choosing a baseline, our primary consideration is to select a method that has an impact; secondly, the baseline should be distributed and have the potential to be deployed in large-scale networks. Based on the above considerations, we have chosen the following three methods as baselines.

1. WAVE [13]: Each time DR receives a join request, it needs to probe all nodes in the multicast tree and select the MJN with the lowest path load to join. The WAVE algorithm is suitable for asymmetric routing and has an impact on many algorithms [14–17];

- 2. Near: DR selects the nearest MJN to join. Many ICN multicast protocols use this method to select MJN [26–29]. This method cannot detect the path load status in the case of asymmetric routing;
- 3. SPT: DR adds the root node. Many IP multicast protocols use this method to select MJN [13,18–20]. This method cannot detect the path load status in the case of asymmetric routing.

Our experiment assumes that each multicast group occupies a bandwidth of 6 Mbps (2K video bandwidth), the bandwidth of each link is 10 Gbps (1500 multicast streams), and the switch can accommodate 1K forwarding status at most. Our experiments are conducted on a Dell R740xd PowerEdge server with two Xeon(R) Gold 5218R CPUs and 300 GB RAM.

In our experiments, the positions of DRs are distributed in nodes with relatively high degrees. This is because DR, as a link device between the internal and external networks, is a core device in a small area and therefore needs to be deployed in nodes with relatively high centrality. The number of DRs accounts for 1% to 0.1% of the total network nodes, which is a common simulation setting for sparse mode multicast [17,35]. Since sparse mode is the main mode for multimedia applications [36–38], this paper simulates under sparse mode.

The value of ζ in Formula (2) varies with changes in topology. For the waxman200 and waxman2000 topologies, since each DR can almost completely explore the topology, the value of ζ has little effect on the results, so we set $\zeta = 1$. In the waxman200k topology, we set ζ to 1.5, and in the waxman2000k topology, we set ζ to 2.

The traffic generation model in the experiment is as follows: for each multicast group, the source DR is randomly selected, and each multicast group has only one source DR. The number of receivers (group size) ranges from 1/20 to 1/3 of the total number of DRs, and their locations are randomly selected from all DRs [17,35]. The specific settings are shown in the Table 2.

Topology	Waxman200	Waxman2000	Waxman200k	Waxman2000k
DR Number	20	200	2000	4000
Group Size	6	20	200	400

Table 2. Traffic generation model.

Additionally, in this experiment, the network routing is asymmetric, meaning that the route from point A to point B is often different from the route from point B to point A.

4.1. The Comparison of the Control Message Numbers

In this section, we compared the average number of control messages required for each algorithm to establish a multicast tree. As the number of control messages increases, the network burden becomes heavier, making this indicator very important. In the RSPT algorithm, the DR sends a request to the leaf nodes every certain period, which is set to send a request every time the DR processes 30 new multicast group members. In the WAVE algorithm, DR needs to detect all nodes in the multicast tree whenever a new group member joins. Therefore, we can anticipate that the control message count in RSPT will be much lower than that of WAVE. In contrast, the Near and SPT methods do not require any control message, so their control message count is zero.

In the left graph of Figure 3, we compared the control message counts of different algorithms in topologies of different sizes. Firstly, as the topology expands, the control message count gradually increases. The reason is that for the WAVE algorithm, as the topology grows, the size of the multicast tree also increases, and DR needs to probe more nodes; thus, the control message count gradually increases. For the RSPT algorithm, the larger the topology, the more nodes covered by DR neighbours, so the control message count also increases accordingly. In the topologies of 200, 2k, and 200k nodes, their diameters are 6, 8, and 13, respectively, and the DR's probing diameter is 6. According to statistics,

in these three topologies, the average number of nodes probed by DR is 134, 623, and 932, respectively. In the topology of 2 million nodes, the DR's probing diameter expands to 4, so the number of probed nodes will increase significantly, with an average of 6789 nodes probed. In summary, for RSPT, the larger the topology, the more nodes DR needs to probe, and the more control messages are required. Secondly, we can see that the control message count of RSPT is about one-tenth to one-hundredth of that of WAVE, and the cost brought by RSPT to the network is much lower than that of WAVE.



Figure 3. Control message number comparison.

In the right graph of Figure 3, we compared the number of control messages for various algorithms in a topology with 2 million nodes. When each multicast group has 200 members, the number of control messages for RSPT is only 6034, while for WAVE it is 80,096, which is 7.5% of WAVE. As the number of group members increases, the number of control messages for WAVE gradually increases, while for RSPT it remains constant, eventually accounting for only about 1% of WAVE. The reason why RSPT can maintain the same number of control messages is that RSPT only detects the neighbourhood of DR, and its detection behaviour is not related to the size of the multicast tree. However, WAVE must detect the entire multicast tree. Therefore, as the number of group members increases and the multicast tree becomes larger, the number of control messages for WAVE will gradually increase.

The control messages of RSPT will not impose a heavy burden on the network. The control message payload of RSPT is relatively small, only about 20 bytes, and the sending period is relatively long. In this experiment, DR triggers a detection only after processing 30 multicast group requests, which takes about 10 s. Therefore, the bandwidth occupied by RSPT's control message is below 1 Kb/s. In contrast, WAVE's bandwidth usage can reach over 100 times that of RSPT, reaching the 1 Mb/s level. When the bandwidth usage of each multicast stream is in the Mb/s range, WAVE's control message bandwidth usage will impose a significant burden on the bandwidth and network devices, while RSPT will not.

In short, RSPT is a cost-effective method.

4.2. Delay Comparison

The delay refers to the time from when DR receives the join request to when it calculates the MJN. We assume a propagation delay of 1ms for each link. The delay of WAVE consists of four parts. The first part is the forwarding delay of the request from the DR to the root node of the multicast tree. The second part is the delay of the request spreading from the root node to the leaf node. The third part is the forwarding delay of the fourth part is the delay in obtaining MJNs from the NMRS. RSPT, Near, and SPT only have a delay in obtaining MJNs from the NMRS. In our experiment, we place the NMRS on the central node.

In the left part of Figure 4, we conducted experiments in topologies of various sizes to compare the delay of different algorithms. It can be seen that the delay of WAVE increases with the topology size. This is because the larger the topology, the larger the multicast tree, and the longer it takes for the DR to complete the detection of all nodes on the multicast tree. In a topology of 200 nodes, the DR needs to spend at least 20 ms to process one join request on average. In a topology of 2 million points, the DR needs to spend 1600 ms. Compared to WAVE, RSPT, SPT, and Near have almost no delay, only a delay in obtaining the MJN list from the NMRS, which is less than 10 ms.



Figure 4. Delay comparison.

In the right part of Figure 4, we conducted experiments in the topology with 2 million points. It can be seen that the delay of WAVE increases with the size of the multicast group. This is because the larger the multicast group, the larger the multicast tree, and the longer it takes for the DR to complete the detection of the multicast tree. However, the delay of RSPT, SPT, and Near are not affected by the group size at all. This indicates that RSPT, SPT, and Near can be applied to scenarios with extremely large group members, while WAVE cannot.

In short, RSPT is a low-latency method.

4.3. Traffic Congestion Comparison

In this experiment, we compared the traffic congestion of various methods. The degree of traffic congestion is an important indicator for evaluating MJN selection methods. Reasonable MJN selection methods can effectively reduce traffic congestion and improve multicast scalability. We use max link stress to measure traffic congestion. Max link stress refers to the number of multicast flows carried on the busiest link in the entire topology, measured in units of "group".

Results are shown in Figure 5, It can be seen that RSPT, like WAVE, can effectively balance the traffic in the network. In the 200 and 2000-node topologies, RSPT and WAVE perform almost identically at certain times, with an average performance difference of less than 5%. This is because the diameter of these two topologies is 5 and 8, respectively, and each DR can explore a topology with a diameter of 6, allowing for the very precise exploration of most of the network. However, the detection of RSPT is not in real-time. In fact, during the experiment process, DR will only initiate a detection and update the cached network state after processing 30 join requests. This delay causes RSPT to not be able to perceive the network state in real-time like WAVE, which results in performance differences between RSPT and WAVE, even in small topologies. In the 200,000-node topology with a diameter of 16, and the 2 million-node topology with a diameter of 34, RSPT can only explore part of the topology with 6 or 8 hops, resulting in performance degradation. The impact of topology size on performance will be further analyzed later.



Figure 5. Link load congestion comparison.

Moreover, in each topology, the performance of RSPT is far superior to SPT and Near. This is because RSPT can effectively detect the load on the multicast path, while Near and SPT cannot. SPT establishes a long forwarding path from the root node to the receiver, which cannot sense the network load status, and puts a heavy burden on the network due to the long multicast path. Although Near selects the nearest MJN, it also causes traffic congestion because it cannot sense the network load status.

To demonstrate the impact of topology size on the performance of RSPT and WAVE, we compare the relative performance gap between the two on topologies of different sizes. Results are shown in Figure 6. To illustrate the gap between RSPT and WAVE, we defined the traffic gap index, that can be expressed by Formula (3).

$$TrafficGap = \frac{ML_{RSPT} - ML_{WAVE}}{ML_{WAVE}}$$
(3)

ML represents the max link stress of the network. TrafficGap refers to the difference in traffic congestion level between RSPT and WAVE. As the traffic congestion level of RSPT has always been higher than that of WAVE, this index is greater than 0. The larger the value of this index, the more serious the traffic congestion of RSPT compared to WAVE, and the worse the performance relative to WAVE. The smaller the value, the more similar the traffic congestion level between RSPT and WAVE, and the closer their performance.

Through Figure 6, we can observe that as the topology scale increases, the TrafficGap index also gradually increases. However, when the topology size reaches 2 million points, the TrafficGap index sharply drops. This is because as the topology grows, the proportion of the neighbourhood that RSPT can detect in the entire topology decreases, leading to the imprecise evaluation of path cost by DR. This increases the traffic load congestion of RSPT and also increases the traffic gap index. However, when the size of the topology reaches 2 million nodes, the detection radius of RSPT changes from 3 to 4 hops, and DR can perceive a larger range of the network, thereby reducing the traffic load congestion and causing a sudden drop in the traffic gap index.

From Figure 6, we can also see that as the number of multicast group members increases, the TrafficGap gradually decreases. This is because as the number of multicast group members increases, the multicast tree becomes larger, allowing DR to perceive the

load status of more multicast tree nodes. This reduces the degree of flow congestion of RSPT, that is, the TrafficGap decreases.



Figure 6. Traffic gap comparison.

Overall, the maximum TrafficGap is around 17%, and in topologies with less than 200,000 nodes, the TrafficGap is below 5%. Therefore, the degree of flow congestion of RSPT and WAVE is quite similar, especially when the topology node number is less than 200,000.

4.4. MFT Congestion Comparison

The level of MFT congestion is another important indicator for measuring the effectiveness of MJN selection methods. A reasonable MJN selection method can effectively reduce MFT congestion, avoid single-point performance bottlenecks, and improve multicast scalability.

In Figure 7, we have illustrated the level of MFT congestion for various algorithms. We use max MFT entry number to measure MFT congestion. The max MFT entry number refers to the number of multicast forwarding states carried by the most heavily loaded node in the entire topology, measured in units of "count".



Figure 7. MFT Congestion comparison.

load. Therefore, the granularity of adjustment for link states are smaller than that of the forwarding states. This results in WAVE or RSPT having a more severe congestion level for forwarding states than for traffic load. Specifically, the congestion level for forwarding states in WAVE and RSPT can reach about 40% of that achieved by SPT, while the congestion level for the traffic load is only about 30% of that achieved by SPT.

The curve of MFT congestion for different algorithms is similar to that of link stress congestion. The performance difference between RSPT and WAVE is within 12%, indicating that RSPT can effectively balance a forwarding state load. Although Near can join the multicast tree using the shortest path, its performance is slightly inferior to RSPT and WAVE due to its inability to avoid high-load nodes. Due to building long paths, SPT can increase the workload of more nodes; At the same time, SPT cannot perceive the network load status. Therefore, SPT has the worst performance.

To demonstrate the impact of topology size on the performance of RSPT and WAVE, we compare the relative performance gap between the two on topologies of different sizes. Results are shown in Figure 8. To illustrate the average difference between RSPT and WAVE, we defined the MFT gap index, which can be expressed by Formula (4).

$$MFTGap = \frac{MT_{RSPT} - MT_{WAVE}}{MT_{WAVE}}$$
(4)

MT represents the max MFT entry number of the network. MFTGap refers to the difference in forwarding state congestion level between RSPT and WAVE. As the forwarding state congestion level of RSPT has always been higher than that of WAVE, this index is greater than 0. The larger the value of this index, the more serious the forwarding state congestion of RSPT compared to WAVE, and the worse the performance relative to WAVE. The smaller the value, the more similar the forwarding state congestion level between RSPT and WAVE, and the closer their performance.



Figure 8. MFT gap comparison.

The conclusion we obtained from Figure 8 is similar to that from Figure 6. The difference lies in that, from an overall perspective, the MFTGap index is around 12%, which is 5% lower than the TrafficGap index. This is because achieving balance in the forwarding state is difficult for both WAVE and RSPT, which we have discussed before. Therefore, the difference between the two is not as significant as when balancing the traffic load.

4.5. Multicast Capacity Comparison

Multicast capacity refers to the number of multicast groups that a network can accommodate. The reason why there is a limit to the number of multicast groups that a network can accommodate is mainly due to the limit of the forwarding state that nodes can accommodate and the limit of the link bandwidth. The experiment will terminate when the network can no longer accommodate more multicast groups. At this point, the number of multicast groups that the network can accommodate is the multicast capacity.

In this experiment, we assume that each node can accommodate up to 1k forwarding states, and each link can accommodate up to 1.5 k multicast flows. This experimental setup is based on commercial switching device, such as the Huawei CloudEngine S16700 series and CloudEngine S8700 series switches [39].

As can be seen from Table 3, SPT has the minimum multicast capacity because it establishes long multicast paths and cannot perceive the network load status, making it easy to cause load congestion. WAVE has the largest multicast capacity because it can detect the network load most accurately. The multicast capacity of RSPT is similar to that of WAVE, reaching 94%, 87%, 82%, and 89% of WAVE performance in four different topologies, respectively. At the same time, the multicast capacity of RSPT is 2.34 times that of SPT and 1.94 times that of Near. The reason why RSPT can have such a similar performance as WAVE and significantly surpass SPT and Near is because it obtains the load status that can effectively detect the multicast path established downlink, thereby avoiding high-load links and nodes.

Table 3. Multicast capacity of each algorithm.

Multicast Capacity	200	2 k	200 k	2000 k
SPT	2000	5421	21,000	29,784
WAVE	3877	11,773	51,747	78,378
RSPT	3541	10,243	42,433	69,765
Near	2720	8310	32,241	43,320

It is worth noting that the performance gap between RSPT and WAVE first increases and then decreases. This is because as the topology increases, RSPT's perception error will become larger, and therefore the performance gap with WAVE will increase. However, in the topology of 2 million points, RSPT's detection range has been increased to 4, so the perception accuracy has been improved, and the gap with WAVE has been reduced. This is in line with our previous analysis.

It can be seen from Table 3 that the larger the network, the greater its multicast capacity. This is easily understandable. With a larger network, there is more room for adjustment in the multicast tree, which is more conducive to avoiding high-load areas. Therefore, larger networks are less likely to experience congestion compared to smaller networks, and can accommodate more multicast groups.

4.6. Simulation Conclusion

By comparing the number of control messages and delays, it can be proved that the method proposed in this article has the advantages of low cost (using fewer control messages) and low latency (almost real-time response). In addition, by comparing the traffic congestion level, the forwarding status congestion level, and multicast capacity, we can find that the proposed method can reduce load congestion and improve multicast capacity.

Among existing methods, Near and SPT are two MJN selection methods that have low processing delays and minimal control messages. RSPT has more than twice the multicast capacity improvement compared to Near and SPT, so it can be chosen as the MJN selection method when a large multicast capacity is required. WAVE has the largest multicast capacity, but it has a large amount of control messages and extremely high processing delays. If deployed in practical engineering environments, it will greatly reduce the quality of user services and bring a heavy burden to the network. Based on this, in practical engineering environments, RSPT should be chosen as the MJN selection method instead of WAVE.

5. Conclusions

One of the challenge of multicast technology is to efficiently establish downlink multicast paths in the case of asymmetric routing. To solve this problem, we propose an MJN selection algorithm based on the reverse shortest path tree (RSPT). DR constructs an RSPT for its k-hop subgraph. The RSPT has the DR as its root and consists of all of the shortest paths from other nodes in the k-hop subgraph to the DR. As long as the leaf nodes of the RSPT send probe messages to the DR, the DR can evaluate the cost of any multicast path in the k-hop subgraph and select the MJN with the lowest cost to join the multicast tree. This method can effectively cope with the establishment of downlink multicast paths (with MJN as the starting point and DR as the endpoint) in the case of asymmetric routing. Compared with existing methods, the proposed method has an extremely low path probing cost and delay, and can effectively balance the network load state, thereby increasing the number of multicast groups accommodated by the network.

Author Contributions: Conceptualization, Z.T., J.Y. and L.H.; methodology, Z.T. and J.Y.; software, Z.T.; writing—original draft preparation, Z.T.; writing—review and editing, Z.T., J.Y. and L.H.; supervision, J.Y. and L.H.; project administration, J.Y. and L.H.; funding acquisition, J.Y. and L.H. 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 is not applicable to this article.

Acknowledgments: We would like to express our gratitude to Xiaoyong Zhu, Ri Han for their support for this work.

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

References

- 1. Latif, A.; Parameswaran, R.; Vishwarupe, S.; Khreishah, A.; Jararweh, Y.; Alenezi, A.H. MediaFlow: Multicast Routing and In-Network Monitoring for Professional Media Production. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 1862–1875. [CrossRef]
- Farshad, A.; Nilsson, M.; Appleby, S. Adaptive media streaming over IP multicast (mABR) a data analytic evaluation. In Proceedings of the 2nd International Workshop on Design, Deployment, and Evaluation of Network-Assisted Video Streaming, Rome, Italy, 9 December 2022.
- 3. González, C.C.; Pizzi, S.; Murroni, M.; Araniti, G. Multicasting over 6G Non-Terrestrial Networks: A Softwarization-based Approach. *IEEE Veh. Technol. Mag.* 2023, *18*, 91–99. [CrossRef]
- Bhattacharjee, S.; Acharya, T.; Bhattacharya, U. Cognitive radio based spectrum sharing models for multicasting in 5G cellular networks: A survey. *Comput. Netw.* 2022, 208, 108870. [CrossRef]
- 5. Harrabi, S.; Jaafar, I.B.; Ghedira, K. Survey on IoV Routing Protocols. Wirel. Pers. Commun. 2023, 128, 791–811. [CrossRef]
- 6. Bousbaa, F.Z.; Kerrache, C.A.; Lagraa, N.; Hussain, R.; Yagoubi, M.B.; Tahari, A.E.K. Group data communication in connected vehicles: A survey. *Veh. Commun.* 2022, *37*, 100518. [CrossRef]
- Cisco.cisco 2022 Annual Report. Available online: https://www.cisco.com/c/en/us/about/annual-reports.html (accessed on 10 February 2023).
- 8. Kamran, R.; Jha, P.; Kiran, S.; Karandikar, A. A Survey on Multicast Broadcast Services in 5G and Beyond. In Proceedings of the 2022 National Conference on Communications (NCC), Mumbai, India, 24–27 May 2022.
- Ahlawat, P.; Tyagi, K. Scalable Rekeying Using Linked LKH Algorithm for Secure Multicast Communication. In Advances in Malware and Data-Driven Network Security; IGI Global: Hershey, PA, USA, 2022; pp. 112–126.
- Diab, K.; Yassini, P.; Hefeeda, M. Orca: Server-assisted Multicast for Datacenter Networks. In Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), Renton, WA, USA, 4–6 April 2022.
- 11. Oliveira, P.; Silva, A.; Valadas, R. HPIM-DM: A fast and reliable dense-mode multicast routing protocol. In *IET Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2023.

- 12. Li, L.S.; Chi, H.I.; Xie, K.C.; Chan, D.Y. Mobility-Sensitive Multicast Protocol in NEMO. *KSII Trans. Internet Inf. Syst.* 2022, 16, 1994–2017.
- 13. Biersack, E.; Nonnenmacher, J. WAVE: A new multicast routing algorithm for static and dynamic multicast groups. In *NOSSDAV*; Springer: Berlin/Heidelberg, Germany, 1995.
- 14. Guler, E.; Karakus, M.; Ayaz, F. Genetic algorithm enabled virtual multicast tree embedding in Software-Defined Networks. *J. Netw. Comput. Appl.* **2023**, 209, 103538. [CrossRef]
- Palmieri, N.; Serianni, A. A proposal of a heuristic QoS multicast routing for multi-layer wireless architecture. In Proceedings of the Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea and Space Vehicles and Infrastructure 2022, Orlando, FL, USA, 3 April–13 June 2022; Volume 12115, pp. 83–91.
- 16. Babu, S.; Parthiban, A.R.K. DTMR: An adaptive distributed tree-based multicast routing protocol for vehicular networks. *Comput. Stand. Interfaces* **2022**, *79*, 103551. [CrossRef]
- Meng, R.; Wan, J.; Xu, B.; Li, F.; Wang, W.; Li, Z.; Wang, C. CGM2: Carrier-Grade Minimalism Multicast with Stateless Explicit Path. In Proceedings of the 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR), Taicang, China, 6–8 June 2022.
- Bhattacharyya, S.; Diot, C.; Giuliano, L.; Rockell, R.; Meylor, J. An Overview of Source-Specific Multicast (SSM). RFC 3569. Available online: https://www.hjp.at/doc/rfc/rfc3569.html (accessed on 17 April 2023).
- Adams, A.; Nicholas, J.; Siadak, W. Protocol Independent Multicast-Dense Mode (PIM-DM): Protocol Specification (Revised). RFC 3973. Available online: https://www.hjp.at/doc/rfc/rfc3973.html (accessed on 17 April 2023).
- Pan, Y.; Lyu, N.; Yang, C. GOR: Group-oblivious multicast routing in airborne tactical networks under uncertainty. J. Netw. Comput. Appl. 2022, 207, 103509. [CrossRef]
- 21. Rafique, W.; Hafid, A.S.; Cherkaoui, S. Complementing IoT Services Using Software Defined Information Centric Networks: A Comprehensive Survey. *IEEE Internet Things J.* **2022**, *9*, 23545–23569. [CrossRef]
- Karim, F.A.; Aman, A.H.M.; Hassan, R.; Nisar, K. A Survey on Information-Centric Networking with Cloud Internet of Things and Artificial Intelligence. *Wirel. Commun. Mob. Comput.* 2022, 2022, 7818712. [CrossRef]
- Nour, B.; Sharif, K.; Li, F.; Biswas, S.; Moungla, H.; Guizani, M.; Wang, Y. A survey of Internet of Things communication using ICN: A use case perspective. *Comput. Commun.* 2019, 142–143, 95–123 [CrossRef]
- Farhan, K.A.; Abdel-Fattah, F.; Altarawneh, F.; Lafi, M. Survey paper on multicast routing in mobile ad-hoc networks. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019.
- 25. Li, B.; Wang, J. An Identifier and Locator Decoupled Multicast Approach (ILDM) Based on ICN. Appl. Sci. 2021, 11, 578. [CrossRef]
- 26. Barakabitze, A.A.; Xiaoheng, T.; Tan, G. A Survey on Naming, Name Resolution and Data Routing in Information Centric Networking (ICN). *Int. J. Adv. Res. Comput. Commun. Eng.* **2014**, *3*, 8322–8330. [CrossRef]
- Li, S.; Zhang, Y.; Raychaudhuri, D.; Ravindran, R. A comparative study of MobilityFirst and NDN based ICN-IoT architectures. In Proceedings of the 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Rhodes, Greece, 18–20 August 2014.
- Fahrianto, F.; Kamiyama, N. Comparison of migration approaches of ICN/NDN on IP networks. In Proceedings of the 2020 Fifth International Conference on Informatics and Computing (ICIC), Gorontalo, Indonesia, 3–4 November 2020.
- 29. Baccelli, E.; Mehlis, C.; Hahm, O.; Schmidt, T.C.; Wählisch, M. Information centric networking in the IoT: Experiments with NDN in the wild. In Proceedings of the 1st ACM Conference on Information-Centric Networking, Paris, France, 24–26 September 2014.
- Fotiou, N.; Nikander, P.; Trossen, D.; Polyzos, G.C. Developing information networking further: From PSIRP to PURSUIT. In Broadband Communications, Networks, and Systems: 7th International ICST Conference, BROADNETS 2010, Athens, Greece, 25–27 October 2010, Revised Selected Papers 7; Springer: Berlin/Heidelberg, Germany, 2012.
- 31. Li, G.; Mishra, D.; Jiang, H. Cooperative NOMA with incremental relaying: Performance analysis and optimization. *IEEE Trans. Veh. Technol.* **2018**, *67*, 11291–11295. [CrossRef]
- García, G.; Pinto-Roa, D.P. Multicast routing and virtual network function placement in NFV-SDN networks: A genetic algorithms approach. In Proceedings of the 2022 Latin America Networking Conference, Armenia, Columbia, 19–20 October 2022; pp. 10–17.
- 33. Qi, X.; Chen, Y.; Jian, R.; Wang, J. Two-level-enforced security with hybrid precoding for multicast massive MIMO wiretap systems. *Phys. Commun.* **2022**, *54*, 101817. [CrossRef]
- Wang, X.; Xing, H.; Song, F.; Luo, S.; Dai, P. Dynamic Multicast-oriented Virtual Network Function Placement with SFC Request Prediction. In Proceedings of the 2022 14th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 10–12 June 2022; pp. 81–88.
- 35. Lakhlef, I.E.; Djamaa, B.; Senouci, M.R. A comprehensive study of multicast routing protocols in the Internet of Things. In Artificial Intelligence and Its Applications: Proceeding of the 2nd International Conference on Artificial Intelligence and Its Applications (2021), Hefei, China, 28–29 October 2021; Springer International Publishing: Cham, Switzerland, 2022.
- Ganesan, E.; Liem, A.T.; Hwang, I.-S. QoS-aware multicast for crowdsourced 360° live streaming in SDN aided NG-EPON. IEEE Access 2022, 10, 9935–9949. [CrossRef]
- Bilal, K.; Shuja, J.; Erbad, A.; Alasmary, W.; Alanazi, E.; Alourani, A. Addressing Challenges of Distance Learning in the Pandemic with Edge Intelligence Enabled Multicast and Caching Solution. *Sensors* 2022, 22, 1092 [CrossRef]

- Luglio, M.; Romano, S.; Roseti, C.; Zampognaro, F. Satellite multi-beam multicast support for an efficient community-based CDN. Comput. Netw. 2022, 217, 109352. [CrossRef]
- 39. Huawei. Available online: https://e.huawei.com/cn/products/switches/campus-switches/s16700 (accessed on 10 February 2023).

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.