A Service Protection Mechanism of Deterministic Networking Based on Segment Routing

Tianchi Li(✉) and Yueping Cai

Chongqing University, Chongqing, China
{20144009,caiyueping}@cqu.edu.cn

Abstract. Deterministic Networking (DetNet) provides guaranteed packet transport services of ultra-low packet loss and bounded delay for the critical traffic in real-time applications such as the industrial control and the power grid. DetNet guarantees reliable packet transmissions by forwarding replicated packets on redundant paths in parallel. This service protection mechanism of DetNet is Packet Replication Elimination and Ordering Functions (PREOF). However, how to obtain the redundant paths and implement the packet replication and elimination functions of the PREOF remains to be a great challenge. This paper proposes an improved PREOF mechanism based on Segment Routing (SR-PREOF). It designs an edge-disjoint path-pair routing algorithm based on the improved Link Pruning method (LP-EDJPP). The proposed SR-PREOF implements the scheme with the SR technology. Network simulation results show that the proposed SR-PREOF effectively improves the packet reception rate and reduces the end-to-end worst-case latency bound while achieving the comparable path reliability performance compared with the traditional PREOF. The packet reception rate of the SR-PREOF increases by 5.6% and the end-to-end worst-case latency bound decreases by 10.89% compared to the PREOF when the offered load is 0.7.

Keywords: Deterministic Networking · Service protection · Segment routing · Routing algorithm

1 Introduction

The rapid development of Industrial Internet of Things brings great changes to the production and life style of human beings, but it presents more strict requirements for network services. Industries such as the industrial automation, control system and power grid have strong demands for *deterministic* network services with bounded delay and ultra-low packet loss [1]. However, the traditional IP network provides users with *best effort* service, which is difficult to meet the strict Quality of Service (QoS) requirements of deterministic traffic on the delay, jitter and packet loss rate. The emergence of Deterministic Networking (DetNet) [2] provides a new solution for the traffic demands of such industries.

F. Fang and F. Shu (Eds.): GameNets 2022, LNICST 457, pp. 83–99, 2022.
https://doi.org/10.1007/978-3-031-23141-4_7

DetNet is a network technology that provides a promised Service Level Agreement (SLA) guarantee for the time-sensitive traffic. It enables the guaranteed QoS services through the mechanisms of the explicit routes, resource allocation and service protection. Moreover, it guarantees the performance of the time-sensitive traffic in terms of the delay jitter, packet loss and end-to-end (E2E) bounded latency.

Three key enabling technologies of DetNet include: 1) **Explicit routes**: Provide explicit paths for the DetNet flows that meet the user's SLA requirements. 2) **Resource allocation:** Book data plane resources for the DetNet flows according to the allocated paths. 3) **Service protection**: Replicate the DetNet flows and transmit the copies in parallel over different paths. The service protection is designed to reduce or eliminate packet loss due to equipment or link failures. The packet loss can be greatly reduced by spreading the data over multiple disjoint paths for transmissions. Traditional network protection methods include $1 + 1$ linear protection [3], network coding [4], PREOF [5] etc. and all of them consume extra network bandwidth resources. The Packet Replication Elimination and Ordering Functions (PREOF) mechanism is proposed in [5] to solve the service protection problems of DetNet. Nevertheless, there are no mature routing algorithms and implementation details of the PREOF. In addition, the efficiency of the method needs to be improved.

Segment Routing (SR) [6] is based on Multi-Protocol Label Switching (MPLS) [7], which has the features of source routing and header instruction programmability. It supports explicit routes on IPv6 or MPLS data plane, which can provide the powerful traffic scheduling capability. This paper proposes an improved PROEF mechanism based on Segment Routing (SR-PREOF). It designs an edge-disjoint path-pair routing algorithm based on the improved Link Pruning [8] method (LP-EDJPP). It can compute the appropriate replication node and elimination node as well as the edge-disjoint path-pair to transmit deterministic traffic flows. The proposed SR-PREOF implements the service protection with the SR technology. Network simulation results show that the proposed SR-PREOF effectively improves the packet reception rate and reduces the E2E worst-case latency bound while achieving the comparable path reliability performance compared with the traditional PREOF. The packet reception rate of the SR-PREOF increases by 5.6% and the E2E worst-case latency bound decreases by 10.89% compared to the PREOF when the offered load is 0.7.

The contributions of this paper are as follows:

1) An edge-disjoint path-pair routing algorithm based on the improved Link Pruning method (LP-EDJPP) is designed considering about the constraints of network topological connections and delays.
2) An improved PREOF mechanism based on Segment Routing (SR-PREOF) is proposed. It utilizes SR technology to implement the scheme for DetNet.

2 Related Work

The whole process of the service protection for DetNet can be described as following steps [2]. First, provide *sequencing information* to the packets of a DetNet Compound Flow (DCF) at the source node. This can be done by adding a *sequence number* or *timestamp* as part of the defined DCF. Second, replicate the packets into multiple DetNet Member Flows (DMFs). It needs an appropriate replication node. Third, send the packets on multiple different paths to the destination node. In this way, when some links of one path fail, the transmissions of packets on the other paths are not affected. Finally, eliminate duplicated packets and order them at an appropriate *elimination node*. However, there is a challenge on finding suitable disjoint paths for packets in DetNet.

Link Pruning [8] is one of the traditional disjoint paths routing algorithms. The steps of Link Pruning include: 1) Find one path with the minimum cost from the given topology diagram. 2) Delete all the links on the path in the topology diagram. 3) Find another path with the minimum cost from the pruned topology diagram. However, this method cannot guarantee the consistency of QoS constraints and even cannot find two disjoint paths sometimes. Besides, other people did many works on disjoint paths routing algorithms. Huong T.T. et al. [9] proposed a global load-balanced routing scheme, which could take advantage of global view of the SDN controller to make a global policy for routing and load balancing. Weiner J. et al. [10] proposed a new method to solve the Maximum Edge Disjoint Paths (MEDP) problem. The proposed method was a heuristic algorithm that built a hybridization of Lagrangian Relaxation and Particle Swarm Optimization. Atallah A. A. et al. [11] designed a disjoint multi-path QoS routing algorithm for E2E networks. Jonatan K. et al. [12] focused on the joint routing and scheduling problem of DetNet. This paper proved that the disjoint routing problem of DetNet is NP-hard and the heuristic algorithm is needed to solve it. Aubry F. et al. [13] proved that SR technology is an approach to realize robustly disjoint paths in Internet Service Providers (ISPs). This paper introduced the theory and model for the robustly disjoint paths. It exploited SR's ability to implicitly specify backup paths. Aubry F. et al. [14] proposed an efficient algorithm that computed K-segmentable disjoint paths with similar latencies. It was possible to provide an $1 + 1$ protection service by using segmentable disjoint paths in an IPv6 network that supports Segment Routing architecture.

The Packet Replication, Elimination and Ordering Functions (PREOF) mechanism was proposed in [5] as a typical solution for the service protection in DetNet. PREOF includes three parts: 1) **In-order delivery**. Packets delivered out-of-order will increase the amount of buffering needed at the destination but also the jitter. The Packet Ordering Function (POF) uses the *sequencing information* to re-order a DetNet flow's packets that are received out-of-order. 2) **Packet replication**. The Packet Replication Function (PRF) replicates these packets into multiple DMFs, and typically sends them along multiple different paths to the destination. 3) **Packet elimination.** The Packet Elimination Function (PEF) eliminates duplicate packets of a DetNet flow based on the *sequencing information* and the history of received packets. In general, PREOF is applied to the DetNet service sub-layer for the packet processing at edge nodes, relay nodes and end systems of DetNet. In the simplest case, each packet is replicated at the source node and transmitted over two disjoint paths to the same destination. However, there are still the following problems to be solved. 1) A method to choose the appropriate replication node and elimination node according to the network topology. 2) An algorithm to compute disjoint path-pair as close in costs as possible. It will help the in-order delivery and lead to less E2E jitter and delay. 3) A programmable implementation deployed on the DetNet data plane.

In this paper, the traditional PREOF is extended and improved. The proposed SR-PREOF mechanism utilizes the LP-EDJPP algorithm to find the appropriate nodes and paths, and implements the service protection in DetNet over IPv6 data plane with the SR technology.

3 Service Protection Mechanism SR-PREOF

The improved PREOF mechanism based on Segment Routing (SR-PREOF) mainly includes three parts: path calculation, SR policy programming and data plane forwarding. Figure 1 shows the schematic diagram of the proposed mechanism. This section shows how the SR-PREOF mechanism can operate within the E2E DetNet domain. The proposed LP-EDJPP algorithm can be implemented in the centralized Path Calculation Module (PCM). In the proposed mechanism, the disjoint path-pair computed by the PCM can be implemented as a sequence of Segments. The SR Policy Programming Module (SRPPM) can encode the path-pair information into the Segment Routing Header (SRH). And it informs the data plane to forward packets according to the SR policy. The Data Plane Forwarding Module (DPFM) forwards packets based on the delivered SR policy.

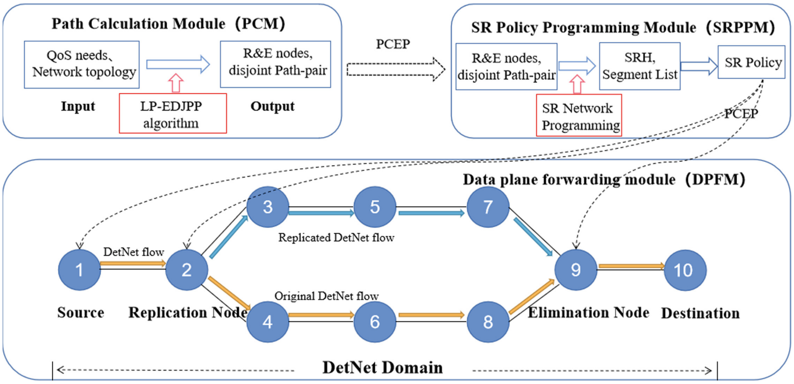

Fig. 1. Schematic diagram of the Deterministic Networking service protection mechanism: SR-PREOF

3.1 Path Calculation

The problem of the path calculation formulated in this paper can be described as follows. Network topology G, source node S and destination node D are given. Computing two disjoint paths between the two nodes and the two paths satisfy a certain objective function. The objective function here is MinSum function and the problem is called MinSum-Balanced 2DP Problem [15]. In this problem, the two disjoint paths we get have the smallest sum of weights and the absolute value difference between the weights of the two paths is the smallest. The problem was proved NP-Completeness in undirected graphs. Therefore, we propose a heuristic algorithm based on the shortest paths approach.

The path calculation module adopts the edge-disjoint path-pair routing algorithm based on the improved Link Pruning (LP-EDJPP) to select edge-disjoint path-pair, the replication node and elimination node for the service protection. The traditional Link Pruning algorithm has following problems: 1) The deletion of links may cause that two disjoint paths cannot be found. 2) The number of links between two disjoint paths may be too different to ensure the consistency of QoS performance constraints. In addition, it may result in out-of-order packet arrivals. To overcome the above problems, we design the LP-EDJPP algorithm. The aim of the LP-EDJPP algorithm is to find the equal-cost edge-disjoint path pairs. The proposed LP-EDJPP algorithm is an approximation algorithm, and the complexity of it is O (N^2). It can get a quality-guaranteed approximate solution in polynomial time.

In this paper, the cost is the latency. In addition, the designed algorithm only considers the edge disjoint (link disjoint) but does not consider the node disjoint. Finding path-pair with completely disjoint nodes and edges will cause excessive delay and bandwidth consumption. The specific process is as follows.

Network Model. The network topology can be represented by an undirected weighted graph $G = \{V, E, C\}$, where V is a set of nodes, E is a set of edges and C is the edge weight set which is a function from E to R^+. This function corresponds to the latencies configured on the edges. Assume that $|V| = N$, $|E| = M$, then the node set $V = \{v_1, v_2, \ldots, v_n\}$, the edge set $E = \{e_1, e_2, \ldots, e_m\}$. The edge weight set $C = \{w_1, w_2, \ldots, w_m\}$. (u, v) represents an edge between node u to node v, where $u, v \in V$. $w = (u, v)$ represents the weight of the edge.

A path P is a sequence $(v_1, v_2, \ldots, v_h)$ such that $(v_i, v_{i+1}) \in E$ for all i and $v_i \neq v_j$ *for* $i \neq j$. $P\ (s, d, G)$ represents the shortest path of node pair (s, d). Path-pair X-Y disjoint means $E(X) \cap E(Y) = \emptyset$.

Shortest Path Calculation. Dijkstra [16] algorithm is the classical shortest path algorithm, which can find the shortest path of any two nodes in the graph. The time complexity is O (N^2). We can compute the shortest path $P = (v_1, v_2, \ldots, v_p)$. Assume that the network topology $G = \{V, E, C\}$ is given as shown in Fig. 2. The shortest path $P\ (1, 9, G) =$ *1-3-6-8-9* between the node pair $(1, 9)$ can be easily calculated according to the Dijkstra algorithm.

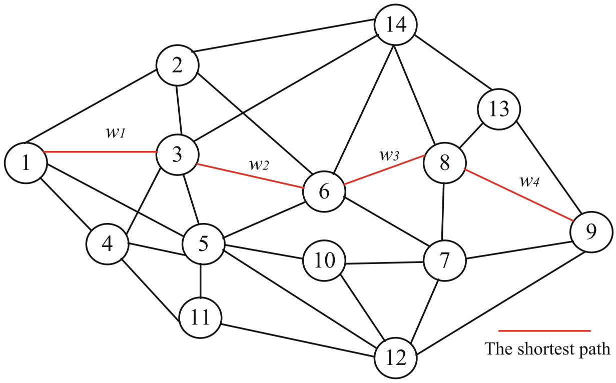

Fig. 2. Schematic diagram of network topology *G* and the shortest path

Edge-Disjoint Path Finding. Set *w* (v_i, v_{i+1}) on path *P* to w*, generate a new network topology G' = (V, E, C'), then use shortest path calculation algorithm to compute another path *P'*. In this case, as shown in Fig. 3, set the weight of each edge on *P (1, 9, G)* to *w**. Note that *w** is a constant, where *w** = *Sum C(P)*. Run the Dijkstra algorithm based on the new network topology, calculate another path *P'* (1, 9, G') = *1-2-6-7-9*.

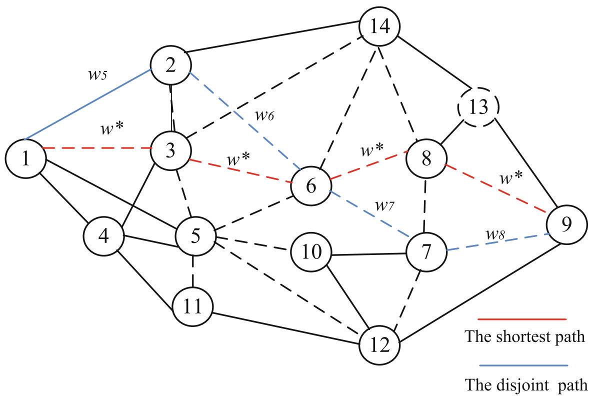

Fig. 3. Schematic diagram of network topology *G'* and the edge-disjoint path

Obtain the Edge-Disjoint Path-Pair. If the two paths are node-disjoint, then the path *P* and *P'* are what we need. Most of the time, we get two paths that have intersection nodes. In this case, we get a new graph G* as shown in Fig. 4. In order to get the equal-cost edge-disjoint path-pair, we need a further optimization. Assume the edge state is *0–1* variables. *0* means unused, *1* means used. The path-pair is X-Y, the edge weight set is W. The optimizing goal is to minimize the latency gap between the path X and Y, as shown in Eq. (1).

$$Min((X - Y) \cdot W) \tag{1}$$

$$s.t.X, Y \in G* \tag{2}$$

We formulate it to an Integer Linear Programming problem, which can be easily solved. In this case, W = {w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8}, and the edge state is: path X = [1, 1, 1, 1, 0, 0, 0, 0], path Y = [0, 0, 0, 0, 1, 1, 1, 1]. Assume that $|(w_1 + w_2 + w_7 + w_8) - (w_3 + w_4 + w_5 + w_6)|$ is the minimum, then we get the edge-disjoint path pair X-Y, where X = 1-3-6-7-9, Y = 1-2-6-8-9.

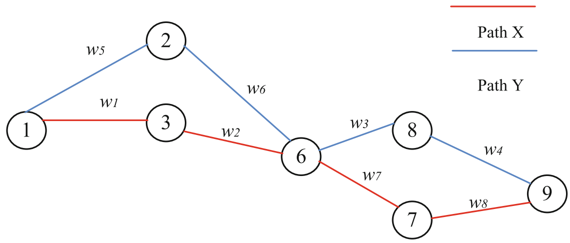

Fig. 4. Schematic diagram of network topology $G*$ and the edge-disjoint path-pair X-Y

The pseudo-code of LP-EDJPP is as follows:

Algorithm 1: Edge-disjoint path-pair routing algorithm based on Link Pruning

```
Input G: network topology
      s, d: source and destination node
Output X, Y: edge-disjoint path-pair
      V={r, e}: replication and elimination node
Start
Dijkstra (s, d, G)→P(s, d, G);// Compute the shortest latency path P from node S to node D
P = (v1,v2,..., vh), set w(vi,vi+1) →w*;
G (V, E, C) →G'(V, E, C');
Dijkstra (s, d, G') →P'(s, d, G'); // Compute another path P'
G*=P+P';
ILP → Min((X-Y) ·W), s.t. X, Y ∈ G*;
Return X, Y; // Solve the ILP equation and get the edge-disjoint path-pair
Explicit X and Y, first shared node → node r, last shared node → node e;
Return V={r, e}
End
```

The path calculation module informs the SR policy programming module of the path calculation results through PCEP [17]. Then, the SR policy programming module encodes the explicit path-pair and service functions into SRH.

3.2 Segment Routing Policy Programming

In the SRv6 [18] network, SR policies are encoded as SRH. When a packet arrives at a node, the node determines how to process the packet based on the semantics associated with the active Segment ID (SID). In SRv6, there are three programming spaces. First, the path information composed of the Segment list can be programmed. Second, the optional TLV fields added in the message can be programmed. The third programmable space is contained in the Segment lists in IPv6 address form. The specific encapsulation format of SRv6 network is shown in Fig. 5.

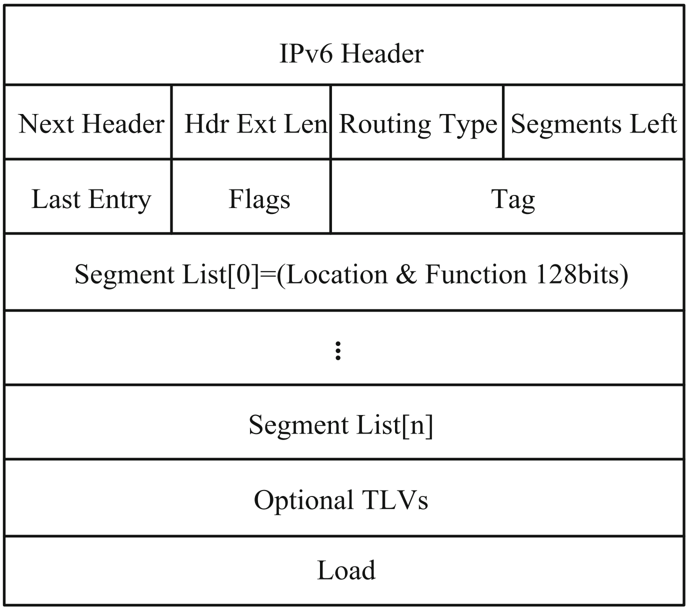

Fig. 5. Schematic diagram of SRv6 network encapsulation format

The Segment list in IPv6 address form (IPv6 SID) consists of three parts: the Locator Segment, the Function Segment and the Argument Segment. The Argument Segment is optional and can be omitted (as shown in the Segment list [0]).

After Locator part is configured, a route will be generated at the local node and diffused out through IGP [19]. Other nodes in the network can locate the node by this route. Therefore, the Locator part has routing function. Function part identifies the instruction bound to the native. If the native receives a SID, Locator part determines first, and if it is published locally, the instruction bound to the native must be processed. Instruction is a native programmable behavior, with different types of instructions. The types of SIDs are also different, different types of SIDs have different functions. The following table briefly introduces several different types of SIDs and their encoded initial registry (Table 1).

The replication and elimination functions are programmed into the Service SID. The *End. B Replication Function* pseudo-code is as follows:

Table 1. Types of SIDs and their encoded initial registry

Type	Value	Hex value	Means
End. SID	1	0×0001	Destination node
End. X SID	5	0×0005	Adjacency
End. DT SID	12	0×0012	Decapsulation

Algorithm 2:End.B Replication Function

Start

1) ***If*** *(NH=SRH & SL>0)* ***Then***
2) ***Extract*** *DetNet TLV values←SRH;*
3) ***Create*** *IPv6-SRH-1,IPv6-SRH-2;*
4) ***Insert*** *SR policy1 & TLV value1→SRH-1,SR policy2 & TLV value2→SRH-2;*
5) ***Delete*** *outer IPv6 SRH header;*
6) ***Create*** *Packet2=Packet1;*
7) ***Encapsulate*** *packet1→ IPv6-SRH-1;*
8) ***Encapsulate*** *packet2→ IPv6-SRH-2;*
9) ***Set*** *IPv6 SA →the local address of this node;*
10) ***Set*** *IPv6 DA of IPv6-SRH-1 →the first segment of SR policy1;*
11) ***Set*** *IPv6 DA of IPv6-SRH-2 →the first segment of SR policy2;*
12) ***ELSE***
13) ***Drop*** *the packet*

End

3.3 Data Plane Forwarding

The data plane forwards packets based on the delivered SR policy. The pseudo-code of SR-PREOF data plane forwarding process is as follows:

Algorithm 3:SR-PREOF data plane forwarding

Start

1) *PCM←DetNet QoS needs (Delay, jitter, bandwidth)*; // The path calculation module obtains deterministic service requirements
2) PCM←*G(V,E); //* PCM gets network topology parameters
3) *LP-EDJPP (s,G,d) →P={X, Y}, V={r, e}*; //Path calculation algorithm
4) *PCM→SRPCM, Create SR policy*; //SRPCM creates SRH
5) *DPFM←SR policy, forward DetNet Flow*;
6) ***Execute*** *End.B. Replication Function in node c*
7) ***Execute*** *End.B. Elimination Function in node e;*
8) ***Forward*** *the DetNet Flow to the DA;*

End

3.4 An Example of SR-PREOF Mechanism

The SR-PREOF mechanism can be simulated by Cisco IOS XR [20]. SRv6 data plane configuration is shown in Fig. 6. The output of path calculation module is replication node R2, elimination node R9, and the explicit path-pair are P1: R2-R3-R5-R7-R9 and P2: R2-R4-R6-R8-R9.

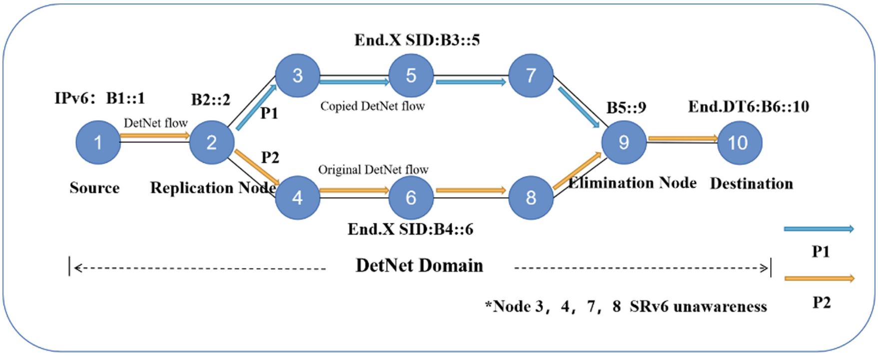

Fig. 6. Schematic diagram of the SRv6 data plane configuration

Then the operation process of the SR-PREOF mechanism is as follows:

SR Policy Encoding at the Source Node. The SR policy programming module encodes the output results of the path calculation as SRH and sends them down to the source node R1, as shown in Fig. 7.

Replication Function at Node R2. The source node sends the DetNet flow to the replication node R2 according to the Segment lists. Node R2 performs a local instruction action based on the outer IPv6 address B2::2 and then performs *End. B Replication Function*. Get two new SRHs: SRH-1 and SRH-2 as shown in Fig. 8.

Forwarding Packets in Parallel. Replicated data packets and original data packets are forwarded explicitly according to the Segment lists in SRH-1 and SRH-2. Note that R3, R4, R7 and R8 nodes without SRv6 function do not handle SRH extension headers, but they lookup IPv6 routing tables for packet forwarding.

Elimination Function at Node R9. Data packets are eliminated when the elimination node R9 hits the *End. B Elimination Function*, which is similar to the replication instruction.

Forwarding Packets to Destination Node R10. After the elimination operation at R9, the new SRH-3 is obtained. The local SID table is searched according to the outer IPv6 address B6::10. Then the *End. X* SID instruction is executed and the traffic is forwarded to the destination node R10.

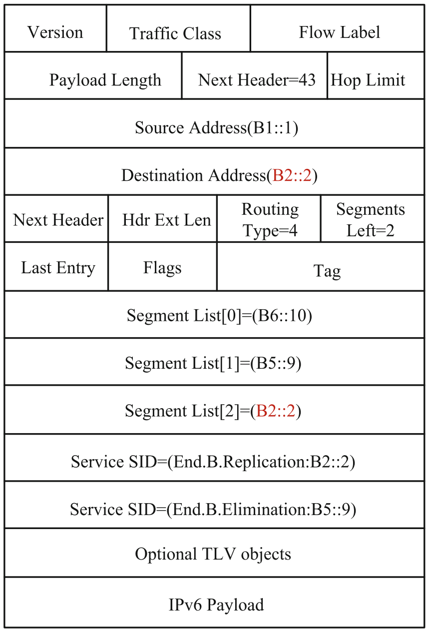

Fig. 7. Schematic diagram of SRv6 packet format of source node R1

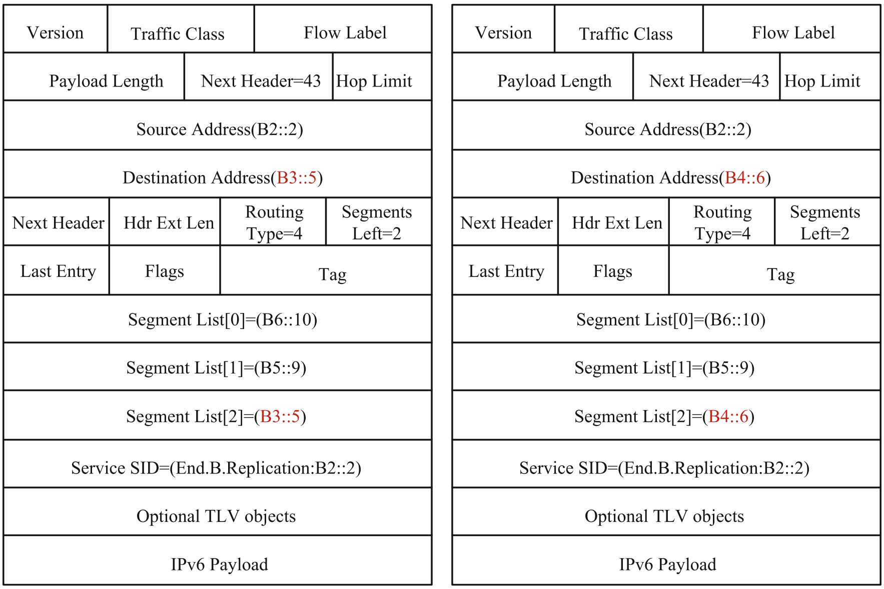

Fig. 8. Schematic diagram of SRH-1 and SRH-2

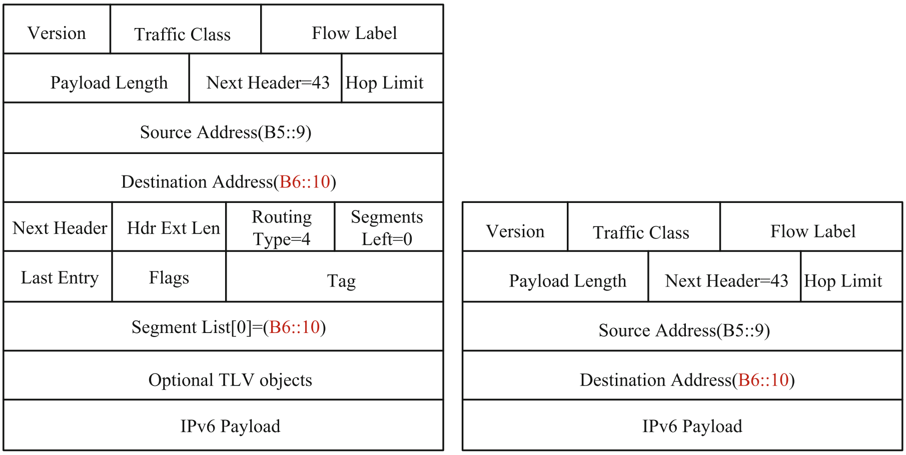

Fig. 9. Schematic diagram of SRv6 packet format of node R9 and R10

Since $SL = zero$, *End. DT6* instruction is executed at node R10, and the SRH extension header is popped to obtain the common IPv6 message, as shown in Fig. 9. The entire DetNet service protection mechanism is finished.

It can be seen from the above example that the service protection in DetNet can be implemented through the SR technology.

4 Performance Evaluation

This section presents a quantitative comparison and analysis of the proposed SR-PREOF mechanism compared with the traditional Shortest Path First (SPF) and PREOF mechanisms. The performance evaluation metrics include the path reliability, packet reception rate and E2E jitter and worst-case latency bound.

4.1 Simulation Settings

The network topology in the simulation is shown in Fig. 10. The edge-disjoint path-pair X-Y is generated according to the designed LP-EDJPP routing algorithm. X = 1-2-3-4-5-6-7, Y = 1-2-8-4-10-6-7. The replication node is node 2 and the elimination node is node 6.

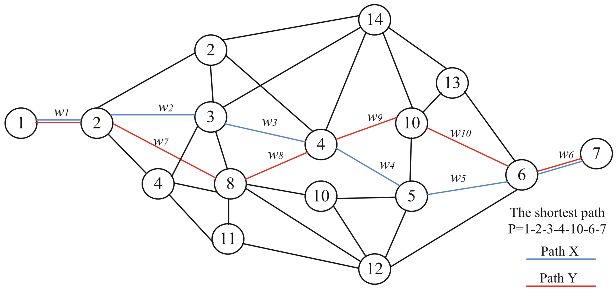

Fig. 10. Schematic diagram of the network topology used in simulation

Table 2 shows the flow model used in the simulation.

Table 2. Flow model and simulation parameters

Parameter	Distribution	Value
Link bandwidth (Gbit/s)	-	1
Probability of edge failure	Uniform distribution	(0, 0.001)
Number of packets	-	5×10^4
Size of packets (bytes)	Uniform distribution	64–500
Arrival process of packets	Poisson distribution	500
Interval of packets (microseconds)	Negative exponential distribution	10

4.2 Performance Evaluation Metrics

Path Reliability. It refers to the probability of successful packet forwarding without edge failures. The path reliability is also associated with the offered load. Equation (3) indicates the probability that each edges in the selected path 1 do not fail. Equation (4) indicates the probability that path 2 successfully forwards the packet. Equation (5) represents the path reliability of the proposed mechanism.

$$R_1 = \prod_{i,j \in V_1} (1 - pij \times offered\_load) \tag{3}$$

$$R_2 = \prod_{i,j \in V_2} (1 - pij \times offered\_load) \tag{4}$$

$$R_t = 1 - (1 - R_1) \times (1 - R_2) \tag{5}$$

where *pij* is the probability of edge failure. The offered load is between (0, 1), which represents the ratio of the current network throughput to the total network bandwidth.

Packet Reception Rate. It refers to the probability that packets can be successfully forwarded to the destination node while meeting the QoS requirements of the service (bounded latency is the requirement in the simulation). In this simulation, the network topology is generated in random with different number of nodes under the same offered load 0.7. Packet reception rate is the ratio of the number of packets arrived within the latency bound over the total number of transmitted packets. We will make a comparison between SR-PREOF, traditional PREOF and traditional Link Pruning (LP) scheme.

End-to-End Jitter and Worst-Case Latency Bound. E2E jitter refers to the variation of packet delay. The deterministic service QoS requirements differ from the traditional requirements in that the deterministic traffic forwarding focuses on the worst-case latency bound, rather than the average latency. We divide the latency into three parts here: 1) Link latency. It is given by the network topology. Latency on each link is constant. 2) Processing latency. It is related to the node computing capacity and the offered load. It is assumed that it follows random distribution from *Ta* to *Tb*. 3) Queuing latency. It occurs at each relay nodes. Its latency bound is 2T, T indicates the cycle time, which is a constant. Equation (6) shows how to calculate latency bound, where *N* means the number of relay nodes.

$$Latency_{bound} = Latency_{link} + N \times (Latency_{processing} + 2T) \quad (6)$$

4.3 Simulation Results and Analysis

Figure 11 shows path reliability comparisons among SR-PREOF, PREOF and SPS mechanisms under different offered loads. With the increase of the offered load, the path reliability of three schemes all decrease. The path reliability of SPS scheme is the lowest, because it does not prepare backup paths. The proposed SR-PREOF can achieve the comparable path reliability performance compared with the traditional PREOF.

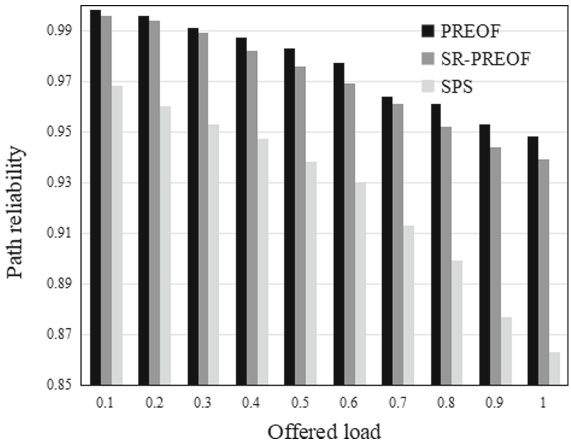

Fig. 11. Path reliability of schemes under different offered loads

As shown in Fig. 12, with the increase of the number of nodes in the network topology, the packet reception rate of the three mechanisms all increase. Moreover, the proposed SR-PREOF always has the highest packet reception rate. The advantage is obvious when the network size is small, because the other two algorithms may not find two disjoint paths. The proposed SR-PREOF increases packet reception rate by 5.6% compared with the traditional PREOF when the number of nodes is 12.

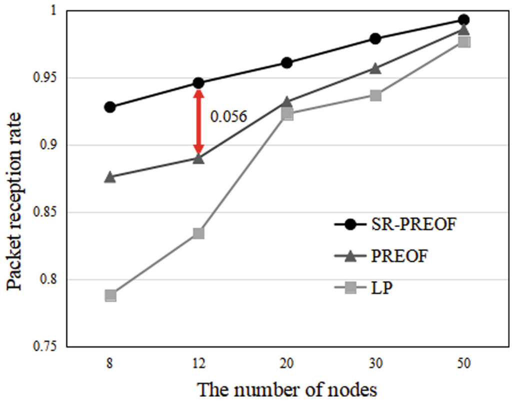

Fig. 12. Packet reception rate under different numbers of nodes

Figure 13 shows the E2E jitter and worst-case latency bound of the proposed SR-PREOF and the traditional PREOF under different offered loads. The SR-PREOF has a better jitter performance than the traditional PREOF. The latency bound of SR-PREOF is more stable than the traditional PREOF. What's more, when the offered load is 0.7, the E2E worst-case latency bound of the proposed SR-PREOF is decreased by 10.89% compared with the traditional PREOF.

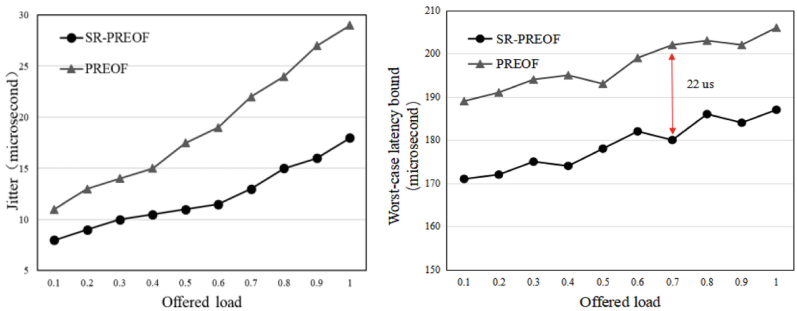

Fig. 13. End-to-end jitter and worst-case latency bound of SR-PREOF and PREOF

5 Conclusion

Based on the Segment Routing technology, this paper proposed the SR-PREOF mechanism for the service protection in DetNet. This mechanism utilizes the designed LP-EDJPP routing algorithm to obtain the equal-cost edge-disjoint path-pair for parallel packet transmissions. Simulation results show that the proposed SR-PREOF effectively improves the packet reception rate and reduces the E2E worst-case latency bound while achieving the comparable path reliability performance compared with the traditional PREOF. The packet reception rate of the SR-PREOF increases by 5.6% and the E2E worst-case latency bound decreases by 10.89% compared to the PREOF when the offered load is 0.7.

References

1. Grossman, E.: Deterministic Networking Use Cases. RFC 8578 (2019). https://www.rfc-editor.org/rfc/pdfrfc/rfc8578.txt.pdf
2. Finn, N., et al.: Deterministic Networking Architecture. RFC 8655 (2019). https://www.rfc-editor.org/rfc/pdfrfc/rfc8655.txt.pdf
3. Sprecher, N., Farrel, A.: MPLS Transport Profile (MPLS-TP) Survivability Framework. RFC 6372 (2011). https://www.rfc-editor.org/info/rfc6372
4. Koetter, R., Medard, M.: An algebraic approach to network coding. IEEE/ACM Trans. Netw. **11**(5), 782–795 (2001)
5. Varga, B., Farkas, J., Berger, L., et al.: Deterministic Networking Data Plane: MPLS. RFC 8964 (2021). https://www.rfc-editor.org/rfc/rfc8964.pdf
6. Filsfils, C., Previdi, S., Ginsberg, B., et al.: Segment Routing Architecture. RFC 8402 (2018). https://www.rfc-editor.org/pdfrfc/rfc8402.txt.pdf
7. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031 (2001). https://www.rfc-editor.org/pdfrfc/rfc3031.txt.pdf
8. Yuchun, G., Fernando, K., Mieghem, P.: Link-disjoint paths for reliable QoS routing. Int. J. Commun. Syst. **16**, 779–798 (2003)
9. Huong, T.T., Khoa, N.D.D., Dung, N.X., Thanh, N.H.: A global multipath load-balanced routing algorithm based on reinforcement learning in SDN. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1336–1341 (2019)
10. Weiner, J., Ernst, A.T., Li, X., et al.: Solving the maximum edge disjoint path problem using a modified Lagrangian particle swarm optimisation hybrid. Eur. J. Oper. Res. 293, 847–862 (2021)
11. Atallah, A.A., Hamad, G.B., Mohamed, O.A.: Routing and scheduling of time-triggered traffic in time-sensitive networks. IEEE Trans. Ind. Inf. **16**(7), 4525–4534 (2020)
12. Krolikowski, J., Martin, S., Medagliani, P., et al.: Joint routing and scheduling for large-scale deterministic IP networks. Comput. Commun. **165**, 33–42 (2021)
13. Aubry, F., Vissicchio, S., Bonaventure, O., Deville, Y.: Robustly disjoint paths with segment routing. In: CoNEXT 2018, 4–7 December 2018, Heraklion, Greece (2018)
14. Aubry, F., Lebrun, D., Deville, Y., Bonaventure, O.: Traffic duplication through segmentable disjoint paths. In: 2015 IFIP Networking Conference, pp. 1–9 (2015)
15. Qi, G.: Research on Two Disjoint Paths (In Chinese). Fudan University (2006)
16. Risald, Mirino, A.E., Suyoto: Best routes selection using Dijkstra and Floyd-Warshall algorithm. In: 2017 11th International Conference on Information & Communication Technology and System (ICTS), pp. 155–158 (2017)

17. Vasseur, J.P., Le Roux, J.L.: Path Computation Element Communication Protocol. RFC 5440 (2009). https://www.rfc-editor.org/pdfrfc/rfc5440.txt.pdf
18. Filsfils, C., Camarillo, P., Leddy, J., et al.: Segment Routing over IPv6 Network Programming. RFC 8986 (2021). https://www.rfc-editor.org/rfc/rfc8986.pdf
19. Gross, P.: Choosing a Common IGP for the IP Internet. RFC 1371 (1992). https://www.rfc-editor.org/pdfrfc/rfc1371.txt.pdf
20. Cisco policies and processes. Cisco Guide to Harden Cisco IOS XR Devices. https://tools.cisco.com/security/center/resources/increase_security_ios_xr_devices.html