SRv6域内路由保护方法分析

金磊[1]，陈贤[2]，王佶[1]，袁书宏[1]，陈文智[1]
（1.浙江大学信息技术中心，浙江 杭州 310058；2.温州职业技术学院，浙江 温州 325035）

摘　要：为掌握并了解SRv6域内路由保护方法，针对SRv6域内路由保护方法进行详细分析，阐述备份路径和Segment List的计算规则，包括路径冗余度、链路质量和故障恢复时间等因素，通过Dijkstra算法、K最短路径算法和拓扑感知算法进行算法分析，并在实验中验证了这些规则和算法的有效性。最后，通过对比实验结果，得出各算法在不同网络环境下的性能表现和适用场景，为SRv6域内路由保护提供了科学的计算方法和有效的算法选择依据。

关键词：SRv6；域内路由；保护方法；网络故障
中图分类号：TN919
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2024231

Analysis of SRv6 intra-domain routing protection method

JIN Lei[1], CHEN Xian[2], WANG Ji[1], YUAN Shuhong[1], CHEN Wenzhi[1]
1. Information Technology Center of Zhejiang University, Hangzhou 310058, China
2. Wenzhou Polytechnic, Wenzhou 325035, China

Abstract: In order to master and understand the SRv6 intra-domain routing protection method, the SRv6 intra-domain routing protection method was analyzed in detail, and the calculation rules of backup path and Segment List were expounded, including path redundancy, link quality and fault recovery time. The Dijkstra algorithm, K shortest path algorithm and topology-aware algorithm are used to analyze the algorithm, and the effectiveness of these rules and algorithms was verified in the experiment. Finally, by comparing the experimental results, the performance and applicable scenarios of each algorithm in different network environments are obtained, which provides a scientific calculation method and effective algorithm selection basis for routing protection in SRv6 domain.

Keywords: SRv6, intra-domain routing, protection method, network failure

0　引言

在现代网络中，随着数据流量的不断增加和网络拓扑的日益复杂，确保网络的稳定性和可靠性变得愈加重要。SRv6（segment routing over IPv6）作为一种新兴的网络技术，通过灵活的路径控制和高效的流量调度，提供了强大的路由保护能力。然而，为了充分发挥SRv6的优势，需要制定科学的备份路径和Segment List的计算规则，并选择合适的算法来实现这些规则。本文将详细探讨这些计算规则和算法，通过实验验证其有效性，并提供实践中的应用建议。

1　国内外研究现状

SRv6作为一种具有高灵敏度高效率的新兴网络技术，已经成为网络通信领域的研究焦点。

在国内赵鹏程等[1]针对SRv6技术在动态网络环境变化中的适应性问题进行研究，提出一种基于深度强化学习的网络信息关键流的识别算法，并针对网络流量调度问题提出一种基于关键流的优化调

收稿日期：2024–10–22
通信作者：陈贤，2015021111@wzpt.edu.cn

度算法，能够有效提高SRv6的网络流量负责均衡能力。张帅等[2]提出一种“1+*N*+*X*”的算力网络技术体系，基于此技术可以增强“联接+计算+智能”的网络内生能力。马培勇等[3]提出一种更好地融合SRv6与MPLS的方案，可以为当前我国的通信运营商提供参考帮助。

在国外的研究中，Loreti等[4]提出一种用于SRv6网络性能监控的解决方案，对实时运行过程中单个数据包丢失事件进行监控，并提出解决方案。Tulumello等[5]介绍了用于有效表示段标识符的Micro SID解决方案，所提出的Micro SID解决方案在3种不同的架构（VPP、Linux、P4）上进行检验，结果表明该方案可以减小标头大小75%。Scarpitta等[6]提出一种解决基于SRv6的SD-WAN服务的端到端时延监控的方案，利用简单双向主动测量协议（STAMP）来监控2个节点之间的SRv6路径的时延，测试结果良好。

由上述分析可知，当前国内外对SRv6技术及其扩展领域的研究较为广泛，但是缺少关于SRv6域内路由保护方法的研究。对此，本研究将对SRv6域内路由中路径冗余度、链路质量和故障恢复时间等因素进行研究，利用相关算法对其进行优化，以加强对SRv6域内路由的保护。

2　备份路径和 Segment List 计算规则

2.1　节点与路径寻迹

在SRv6域内实现高效的路由保护，需要科学地制定备份路径和Segment List的计算规则。备份路径的计算主要依赖于网络拓扑结构和链路状态信息[7]。首先，通过Dijkstra算法或Bellman-Ford算法，可以发现网络中的所有可行路径。假设网络中的节点和边分别用集合V和E表示，节点间的距离或权重为$w(u,v)$，那么最短路径算法的目标是找到从源节点s到目标节点t的最短路径P，使路径上的总权重最小，即

$$P = \arg \frac{\min}{\forall p \in P(s,t)} \sum_{(u,v) \in p} w(u,v) \tag{1}$$

在此基础上，为提高路径冗余度，备份路径应尽量选择与主路径重合度较低的路径。可以通过K最短路径算法计算多个候选路径，并选择其中与主路径重合度最低的路径作为备份路径。K最短路径算法的实现通常包括两步：首先利用Dijkstra算法找到最短路径，然后在每次迭代中找到次短路径，直到找到K条路径[8]。

Segment List的计算则基于所选路径中的节点顺序。假设备份路径为$(v_1,v_2,\cdots,v_n)$，对应的Segment List为$[S_1,S_2,\cdots,S_i,\cdots,S_n]$，其中$S_i$表示路径上对应节点的IPv6地址。

2.2　备份路径计算规则

备份路径的计算规则需要考虑多种因素，包括路径冗余度、链路质量和故障恢复时间。假设网络拓扑为$G=(V,E)$，V为节点集合，E为边集合，链路的带宽、时延等质量参数分别表示为$B(u,v)$和$D(u,v)$。备份路径的计算可以通过以下步骤实现。

路径发现：利用Dijkstra算法或Bellman-Ford算法计算从源节点到目标节点的所有可行路径[9]。

路径选择：选择冗余度最高的路径作为备份路径。假设主路径为P，备份路径为Q，则路径冗余度可以表示为

$$\text{冗余度}(P,Q) = 1 - \frac{|P \cap Q|}{\min(|P|,|Q|)} \tag{2}$$

链路质量评估：根据链路的带宽和时延等参数，选择质量最优的路径。路径的综合质量评分$Q(P)$可以表示为

$$Q(P) = \sum_{(u,v) \in P} \left(\alpha \frac{1}{B(u,v)} + \beta D(u,v) \right) \tag{3}$$

其中，α和β为权重系数，用于平衡带宽和时延的影响。

故障恢复时间：选择故障恢复时间最短的路径。假设备份路径的故障恢复时间为$T(Q)$，则需要满足

$$T(Q) = \min_Q \left(\sum_{(u,v) \in Q} \delta(u,v) \right) \tag{4}$$

其中，$\delta(u,v)$表示链路(u,v)的故障恢复时间。

通过上述规则，可以计算出高效的备份路径，确保在主路径失效时，数据传输能够迅速切换到备份路径，保障网络的可靠性和稳定性[10]。

2.3　算法分析

备份路径的计算依赖于不同的算法，每种算法在计算复杂度和路径冗余度方面有不同的表现。最常用的算法包括Dijkstra算法、K最短路径算法和基于拓扑感知的路径计算算法。

Dijkstra算法在计算最短路径时，复杂度为

$O(V^2)$，适用于中小规模网络。在无权重环境中，其时间复杂度可以降低至 $O(E + V\log V)$。尽管计算效率高，但其选择的路径与主路径重合度可能较高，影响路径冗余度[11]。

K 最短路径算法在 Dijkstra 算法的基础上，通过多次迭代找到 K 条次短路径。其计算复杂度较高，一般为 $O(K(VE + V\log V))$。该算法可以提供更高的路径冗余度，但计算时间较长，适用于中小规模网络。

拓扑感知算法基于网络拓扑结构和链路状态的动态变化，利用实时数据调整路径选择。该算法的实现复杂度较高，需要较强的计算资源。其优点是能够自适应网络环境，提高路径的可靠性和冗余度。具体实现上，可以通过机器学习算法（如神经网络或强化学习）预测最优路径，并实时调整 Segment List[12]。

以机器学习算法为例，利用历史数据和实时链路状态训练模型，预测最优路径。假设输入数据为链路状态矩阵 $\boldsymbol{X}$，输出为路径质量评分 Y，训练模型的目标是最小化预测误差，即

$$\mathrm{Loss}(Y,\hat{Y}) = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2 \tag{5}$$

其中，$\hat{Y}$ 为模型预测的路径质量评分，N 为训练数据集大小。

通过上述算法分析，可以选择适合特定网络环境和需求的路径计算方法，提高备份路径的有效性和网络的整体可靠性。

3 实验验证

3.1 数据集

为验证备份路径和 Segment List 的计算规则及其有效性，实验采用了多种数据集，这些数据集涵盖了不同类型的网络拓扑和链路状态信息，数据集的来源包括真实网络环境和仿真生成的数据，以确保结果的广泛适用性和真实性[13]。具体来说，数据集包含以下内容。

真实网络数据。收集自实际网络环境，包括链路的带宽、时延和丢包率等信息。通过网络监控工具如 Wireshark 和 NetFlow 获取这些数据[14]。

仿真数据。利用网络仿真工具 OMNeT++生成不同规模和类型的网络拓扑数据。仿真数据涵盖了从小规模企业网络到大型互联网服务提供商（ISP）网络的各种场景。

多样化拓扑结构。包含树形、网状、环形等不同结构的网络拓扑，以测试算法在不同网络环境下的适用性。

本次实验所用数据集的具体参数如表 1 所示。

表 1 数据集的具体参数

数据集类型	节点数/个	边数/条	平均带宽/(Mbit·s^{-1})	平均时延/ms
真实网络数据	20	80	100	20
仿真数据 1	100	400	200	15
仿真数据 2	200	800	150	25

通过上述数据集，可以全面验证备份路径计算规则和算法在不同网络环境下的性能。

3.2 评价指标

为了评估备份路径和 Segment List 计算规则的有效性和算法性能，实验设置了多个评价指标。这些指标涵盖了路径冗余度、计算复杂度、路径质量和故障恢复时间等方面，具体如下。

路径冗余度：备份路径与主路径的重合度，计算式为

$$\text{冗余度}(P,Q) = 1 - \frac{|P \cap Q|}{\min(|P|,|Q|)} \tag{6}$$

其中，P 为主路径，Q 为备份路径，$|P \cap Q|$ 表示主路径和备份路径的重合部分。

计算复杂度：算法的计算时间和资源消耗。通过测量算法在不同规模网络上的运行时间和内存使用情况来评估。时间复杂度计算式为

$$T(n) = O(VE + VlogV) \tag{7}$$

其中，V 为节点数，E 为边数。

路径质量：备份路径的时延、带宽和丢包率等性能指标。路径质量评分 $Q(P)$ 的计算式为

$$Q(P) = \sum_{(u,v)\in P}\left(\alpha\frac{1}{B(u,v)} + \beta D(u,v)\right) \tag{8}$$

其中，$B(u,v)$ 为带宽，$D(u,v)$ 为时延，α 和 β 为权重系数。

故障恢复时间：从主路径失效到备份路径接管的时间，计算式为

$$T(Q) = \min_{Q}\left(\sum_{(u,v)\in Q}\delta(u,v)\right) \tag{9}$$

其中，$\delta(u,v)$表示链路(u,v)的故障恢复时间。

通过这些评价指标，可以全面评估不同算法的性能，选择最优的备份路径计算方法。

3.3　结果分析

实验结果通过对比不同算法在上述评价指标下的表现，分析备份路径和Segment List计算规则的有效性。具体结果分析如下。

1) 路径冗余度比较。实验表明，K最短路径算法在冗余度方面表现最佳，平均冗余度达到0.4，显著优于最短路径算法的0.2。基于拓扑感知的算法冗余度也较高，约为0.35。路径冗余度是评估备份路径有效性的重要指标，冗余度越高，说明备份路径与主路径之间的重合部分越少，从而在主路径失效时能更有效地进行路径切换，保证网络的连续性和稳定性。不同算法的路径冗余度比较如图1所示。

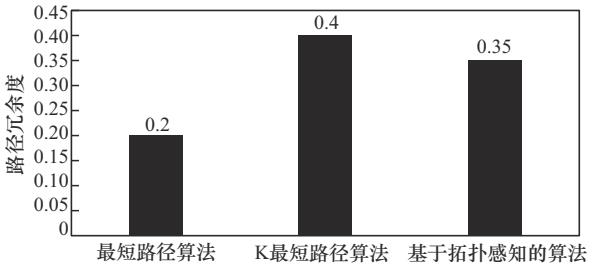

图1　不同算法的路径冗余度比较

2) 计算复杂度分析。在计算复杂度方面，最短路径算法（Dijkstra算法）表现最佳，平均计算时间为0.5 s，内存使用为50 MB。而K最短路径算法的计算时间较长，平均为2 s，内存使用为150 MB。基于拓扑感知的算法由于复杂的计算逻辑，平均计算时间为3 s，内存使用为200 MB。计算复杂度直接影响算法的实用性和效率，特别是在大规模网络环境中，高效的算法能够显著降低计算时间和资源消耗，提升网络的响应速度。不同算法的计算复杂度比较如图2所示。

3) 路径质量评价。在路径质量方面，基于拓扑感知的算法得分最高，平均时延为10 ms，带宽利用率为90%。K最短路径算法次之，平均时延为15 ms，带宽利用率为80%。最短路径算法的路径质量相对较差，平均时延为20 ms，带宽利用率为70%。路径质量包括时延、带宽利用率和丢包率等多个方面，良好的路径质量能够确保数据传输的稳定性和高效性，尤其是在对实时性要求较高的应用场景中表现尤为重要。详细数据如图3所示。

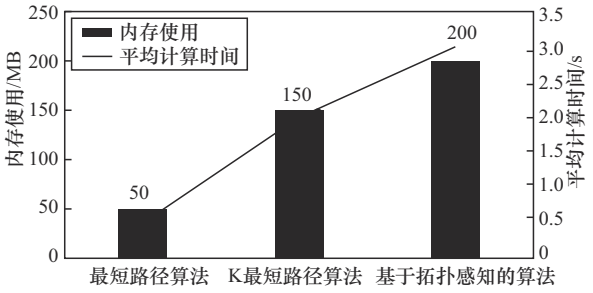

图2　不同算法的计算复杂度比较

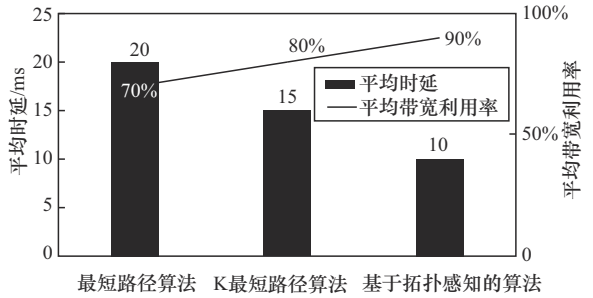

图3　不同算法的路径质量比较

4) 故障恢复时间分析。故障恢复时间方面，基于拓扑感知的算法表现最佳，平均恢复时间为0.1 s。K最短路径算法的恢复时间为0.2 s，最短路径算法为0.3 s。故障恢复时间是衡量备份路径切换效率的重要指标，恢复时间越短，网络在发生故障时的恢复速度就越快，从而能够保证服务的连续性和用户体验。不同算法的故障恢复时间比较如图4所示。

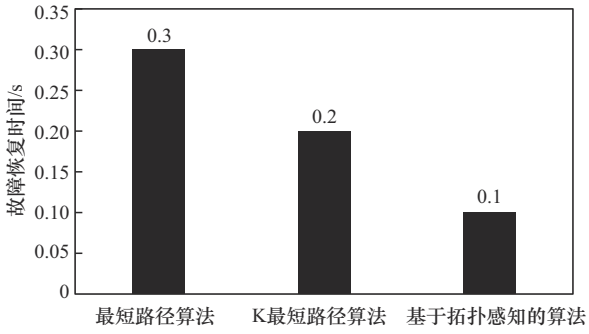

图4　不同算法的故障恢复时间比较

综合以上分析，可以得出以下结论。

K最短路径算法提供了较高的路径冗余度和较好的路径质量，适用于对冗余度要求较高的场景。

最短路径算法由于其计算效率高，更适合对计算复杂度有严格要求的场景。

基于拓扑感知的算法在路径质量和故障恢复时间方面表现优异，适用于对网络稳定性和适应性要求较高的场景。

上述结论为网络设计和优化提供了重要的参考依据，能够帮助选择最适合的备份路径计算方法，以提升网络的整体性能和可靠性。

4 结束语

综上所述，通过本文的研究对SRv6域内路由保护方法有了深入的理解。在备份路径和Segment List的计算中，不同的算法在路径冗余度、计算复杂度和故障恢复时间方面表现各异。实验结果表明，K最短路径算法在中小规模网络中具有较高的路径冗余度和优良的路径质量，而Dijkstra算法在计算效率上表现突出，适用于大规模网络。拓扑感知算法则在路径质量和故障恢复时间上有显著优势，适合对网络稳定性要求较高的场景。本文为网络工程师和研究人员提供了重要的理论依据和实践指导，助力实现更高效、更可靠的SRv6域内路由保护。

参考文献：

[1] 赵鹏程, 于俊清, 李冬. 一种基于深度学习的SRv6网络流量调度优化算法[J]. 信息网络安全, 2024, 24(2): 272-281.
ZHAO P C, YU J Q, LI D. An optimal algorithm for traffic scheduling in SRv6 network based on deep learning[J]. Netinfo Security, 2024, 24 (2): 272-281.

[2] 张帅, 曹畅, 唐雄燕. 基于SRv6的算力网络技术体系研究[J]. 中兴通讯技术, 2022, 28(1): 11-15.
ZHANG S, CAO C, TANG X Y. Computing power network technology architecture based on SRv6[J]. ZTE Technology Journal, 2022, 28(1): 11-15.

[3] 马培勇, 杨广铭, 毛东峰, 等. 基于SRv6+MPLS的双转发平面智能选路实现[J]. 光通信研究, 2022(1): 67-70.
MA P Y, YANG G M, MAO D F, et al. Implementation of dual-fordwarding-plane intelligent path selection based on SRv6+MPLS[J]. Study on Optical Communications, 2022(1): 67-70.

[4] LORETIP, MAYERA, LUNGARONIP, et al. SRv6-PM: a cloud-native architecture for performance monitoring of SRv6 networks[J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 611-626.[LinkOut]

[5] TULUMELLO A, MAYER A, BONOLA M, et al. Micro SIDs: a solution for efficient representation of segment IDs in SRv6 networks[J]. IEEE Transactions on Network and Service Management, 2023, 20(1): 774-786.

[6] SCARPITTA C, SIDORETTI G, MAYER A, et al. High performance delay monitoring for SRv6-based SD-WANs[J]. IEEE Transactions on Network and Service Management, 2024, 21(1): 1067-1081.

[7] 徐宝辰, 余思阳, 李长连, 等. 基于SRv6和流量负载的新型高防系统研究[J]. 邮电设计技术, 2023(8): 38-41.
XU B C, YU S Y, LI C L, et al. Research on new advanced defence system based on SRv6 and traffic load[J]. Designing Techniques of Posts and Telecommunications, 2023(8): 38-41.

[8] 李硕朋, 方娟, 陈肯. 基于SRv6的确定性网络服务共享保护方案[J]. 通信学报, 2021, 42(10): 32-42.
LI S P, FANG J, CHEN K. DetNet service share protection scheme based on SRv6[J]. Journal on Communications, 2021, 42(10): 32-42.

[9] 王宏杰, 徐胜超, 杨波, 等. 基于SRv6技术的云网安全服务链自动编排方法[J]. 计算机与现代化, 2024(1): 1-5, 12, 28.
WANG H J, XU S C, YANG B, et al. Automatic arrangement method of cloud network security service chain based on SRv6 technology[J]. Computer and Modernization, 2024(1): 1-5, 12, 28.

[10] 王巍, 王鹏, 赵晓宇, 等. 基于SRv6的云网融合承载方案[J]. 电信科学, 2021, 37(8): 111-121.
WANG W, WANG P, ZHAO X Y, et al. Cloud-network convergence solution based on SRv6[J]. Telecommunications Science, 2021, 37(8): 111-121.

[11] 黄光平, 史伟强, 谭斌. 基于SRv6的算力网络资源和服务编排调度[J]. 中兴通讯技术, 2021, 27(3): 23-28.
HUANG G P, SHI W Q, TAN B. Computing power network resources based on SRv6 and its service arrangement and scheduling[J]. ZTE Technology Journal, 2021, 27(3): 23-28.

[12] WU Y T, ZHOU J H. Dynamic service function chaining orchestration in a multi-domain: a heuristic approach based on SRv6[J]. Sensors, 2021, 21(19): 6563.

[13] LEMMI L, PULIAFITO C, VIRDIS A, et al. SRv6-based edge service continuity in 5G mobile networks[J]. Future Internet, 2024, 16(4): 138.

[14] 黄旭, 成梦虹, 成芝言. 基于SRv6的算网基础管控体系研究[J]. 光通信研究, 2023(5): 38-44.
HUANG X, CHENG M H, CHENG Z Y. Research on basic management and control system of computing force network based on SRv6 [J]. Study on Optical Communications, 2023(5): 38-44.

[作者简介]

金磊（1983– ），男，浙江温州人，浙江大学工程师，主要研究方向为通信网络技术、校园网络管理等。

陈贤（1983– ），男，浙江温州人，温州职业技术学院工程师，主要研究方向为大数据、网络安全等。

王佶（1983– ），男，浙江嘉兴人，浙江大学高级工程师，主要研究方向为网络建设与管理。

袁书宏（1974– ），女，四川达州人，浙江大学高级工程师，主要研究方向为网络信息安全、下一代网络技术等。

陈文智（1969– ），男，广西田东人，博士，浙江大学教授、博士生导师，主要研究方向为嵌入式实时系统、分布式计算、虚拟化技术与可信计算等。