电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

DOCTORAL DISSERTATION

论文题目 利用图分割的多步网络故障检测技术

学科专业 通信与信息系统

学 号 **201821010306**

作者姓名 陈 偲

指导教师 许 都 教 授

分类号 密级

UDC [注1]

学 位 论 文

利用图分割的多步网络故障检测技术

陈 偲

（作者姓名）

指导教师 许 都 教 授

电子科技大学 成 都

（姓名、职称、单位名称）

申请学位级别 硕士 学科专业 通信与信息系统

提交论文日期 2021.03.19 论文答辩日期 2021.05

学位授予单位和日期 电子科技大学 2021 年 6 月

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

Research of Mutil-stage Network Fault Detection by Graph Partition

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline: Communication and Information System

Author: Chen Cai

Supervisor: Prof. Xu Du

School: School of Information & Communication Engineering

摘　要

日益增长的网络规模对网络的故障探测和定位提出了更高的要求，网络测量是发现网络故障的重要手段。主动测量由于具有灵活性和隐私性在网络故障检测中被广泛运用，但主动测量的方法会向网络中引入额外的探测流量。本研究的测量对象是网络中的链路级故障。为了探测到所有链路故障，最常见的设计是对链路进行全覆盖的测量，这无疑会引入大量的探测流量。而在任何的典型场景中，故障链路的数量都只是网络中的一小部分。理想的设计目标是使用最少的探测来覆盖这些少数的故障链路。为了减小主动测量产生的测量代价，本文设计了一种基于图分割的两阶段测量方法，主要工作如下：

（1） 提出了基于分区的链路传输重要度评价方法，给出了该算法的示例以及和其他重要度评价方法的对比。然后按重要度排序筛选出一轮测量的待测链路集。根据待测链路集设计了基于贪心的探测路径选择算法。对一轮测量进行了仿真建模，验证了能够通过较少待测链路探测到大部分故障的假设，并且讨论了分区数对测量的影响。

（2） 根据一轮测量返回的探测路径状态，二轮测量需要找到可疑区域，并对可疑区域进行进一步的探测。为了得到可疑区域，对一轮探测的探测路径设计了一种基于链路状态的故障定位算法，并提出了最小可识别集的概念。定位得到的最小可识别集被称为最小故障集，其对应的区域称为可疑区域。针对可疑区域中未被一轮探测路径覆盖的链路，采用基于全覆盖思想的二轮测量的探测路径选择算法得到二轮探测路径。

（3） 为了对本文提出的测量方法产生的测量代价进行评估，选择了三类常见的随机网络模型，选择不同参数生成了各个规模的实验拓扑。为了充分验证方法的适用性，选择了三类常见的流量模式对实验拓扑进行了仿真实验。实验证明，在保持较高故障检出率的前提下，对比于常见的全覆盖方法，本文提出的基于分区的测量方案在测量代价上具有显著优势。

关键词：故障检测；测量代价；链路重要度；图分割

ABSTRACT

The ever-increasing network scale puts forward higher requirements for network fault management, and network measurement is the most important method to discover network anomalies. Active measurement is widely used in network anomaly detection due to its flexibility and privacy, but the active measurement method will inject additional detection traffic into the network. The measurement object of this research is the link-level failures. To detect all link failures, the most common design is the measurement cover all links, which will undoubtedly inject a large number of detection packets. In any typical scenario, the number of congested links is only a small part of the network. The ideal design goal is to use the fewest probes to cover these few congested links. To reduce the measurement cost caused by active measurement, this paper designs a two-stage measurement method based on graph partition. The main work is as follows:

(1) We proposed a partition-based link transmission importance evaluation method and given an example of the algorithm. Then we could get a set of links to be tested for the first round of measurement according to the order of importance. A greedy-based algorithm is designed to select detection paths according to the set of links. A simulation model of the first round measurement was performed to verify that this method can detect most of the faults, and we also verified the influence of the number of parts on the measurement.

(2) According to the detection path status returned by the first round of measurement, the second round of measurement needs to find the suspicious area and further detect the suspicious area. To obtain the suspicious area, we proposed a fault location algorithm based on the link-state and a concept of the minimum recognizable set. The minimum identifiable set obtained by the positioning is called the minimum fault set, and the corresponding area is called the suspicious area. The second round measurement aims at the links in the suspicious area that are not covered by the detect paths selected in the first round.

(3) To evaluate the measurement cost of the measurement method proposed in this paper, three types of common random network models are selected, and different parameters are chosen to generate experimental topologies of various scales. To thoroughly verify the applicability of the method, three types of common traffic patterns

were selected to simulate the experimental topology. Experiments have proved that, compared with common full coverage methods, the partition-based measurement scheme proposed in our paper has significant advantages in measurement cost.

Keywords: Fault detection; measurement cost; link importance; graph partition

目 录

第一章 绪 论 1
1.1 研究工作的背景与意义 1
1.2 本文的主要贡献与创新 2
1.3 本论文的结构安排 2
第二章 相关工作 4
2.1 网络测量 4
2.1.1 网络测量性能指标 4
2.1.2 主动测量技术 5
2.1.3 被动测量技术 6
2.2 节点和边的重要性 6
2.2.1 节点重要性 7
2.2.2 边重要性 8
2.3 图分割算法 8
2.3.1 图分割概念 9
2.3.2 图分割相关研究 9
2.4 故障检测与定位 12
2.4.1 故障检测探针 13
2.4.2 故障定位探针 14
2.5 本章小结 15
第三章 基于分区传输重要度排序的待测链路选择 16
3.1 设计思路 16
3.2 拓扑图分割 18
3.2.1 分区方案和全覆盖方案的对比 18
3.2.2 图分割 19
3.3 基于分区的链路传输重要性评价方法 23
3.3.1 问题描述 23
3.3.2 基于分区的传输重要性评价方法 24
3.3.3 算法示例与对比 26
3.4 基于贪心的探针选择算法 28
3.4.1 问题描述 28

3.4.2 基于待测链路集的探针选择算法 29
3.5 仿真分析 30
3.5.1 对比方案和性能指标 30
3.5.2 仿真分析 30
3.6 本章小结 34
第四章 可疑区域的定位和探测 35
4.1 问题描述 35
4.2 可疑区域定位 35
4.2.1 问题建模 35
4.2.2 故障路径中的故障链路定位 36
4.3 可疑区域的探测 40
4.3.1 测量代价分析 40
4.3.2 可疑区域分析 42
4.3.3 二轮探测路径选择 43
4.4 本章小结 44
第五章 多步分区主动测量方法的仿真实验和结果分析 45
5.1 拓扑模型 45
5.1.1ER 随机网络 45
5.1.2 小世界网络 46
5.1.3 无标度网络 46
5.2 流量模式 47
5.3 仿真分析 47
5.3.1 实验拓扑 48
5.3.2 仿真方法介绍 48
5.3.3 不同待测链路比例下的仿真结果分析 51
5.3.4 考虑额外链路信息的待测路径选择 58
5.4 本章小结 60
第六章 总结和展望 61
6.1 本文工作总结 61
6.2 后续工作展望 61
致 谢 63
参考文献 64
攻读硕士学位期间取得的成果 71

第一章 绪 论

1.1 研究工作的背景与意义

这些年来，通信网络在规模和复杂性上都有了显著的增长。复杂性的增加对网络故障检测提出了严峻的挑战。在大型通信网络中，故障是不可避免的，及时检测和识别故障对网络的可靠运行至关重要。网络测量技术最重要的作用之一是识别和定位网络故障的存在。准确的故障检测和诊断对网络应用的稳定性、一致性和性能具有重要意义。故障是网络事件，是网络中可能发生的问题的根本原因。网络中一个组件的故障可能会导致其他组件的故障，故障可能发生在硬件设备(如路由器、链接)中或软件(如路由表中的故障、失败的 web 服务)中。症状是网络故障的外在表现，可以通过观察测量网络状态来分析故障的位置和成因。

主动探测是网络测量的一种主要方法，也被广泛运用于网络的故障检测。主动探测技术能够快速、准确地检测出故障的发生。但是这些方法受到流量开销的限制，特别是在大规模网络中。现有对故障的检测通常建立在对节点或链路的全覆盖上，这样做的好处是只要网络中存在故障就能够被发现，不容易漏检，但主动探测探测方法会消耗网络的带宽资源，全覆盖的方案不可避免的会引入较多的探测流量，目前大多数研究都集中在探测调度方法上，以减少带宽消耗。

网络拓扑通常被抽象为图，在网络规模迅猛增长的当今社会，由网络拓扑获得的图数据的规模也在不断扩大。将大规模图数据进行划分处理是减小图的处理和计算代价的有效方法。如果能将图分割的思想运用于网络测量，能否减小网络的测量代价，这是一个可以思考的方向。

网络中的故障呈现出“二八分布”的特点，即 80%的故障发生在 20%的设备中。如果能够找到那 20%故障源并对其进行测量，可以极大的减小故障检测的代价。在链路级的故障检测中，如何找出这部分的链路，选择的标准又是什么，这也是一个值得探讨的问题。

本文的研究着眼于以上两个思考，主要研究能否利用图分区的思想和通过部分链路的测量实现较少的测量代价，同时保证一定的故障检出率。本文提出了基于分区的链路传输重要度评价算法，利用重要度排序筛选待测链路集，通过的待测链路集的链路状态判断其所在分区的状态，以进行进一步的测量。

1.2 本文的主要贡献与创新

本论文以多路径复杂网络为背景，探讨了基于分区的传输重要度评价方法，对具体的探测路径选择和故障的定位算法进行了研究，并对所提出探测方法的测量代价进行了仿真和对比。本文的测量分为两轮，在第一轮测量中向根据重要度选出的探测路径发送探针并返回测量结果，第二轮测量通过一轮测量的结果对发生故障的可疑区域进行进一步的测量

本文主要创新点与贡献如下：

（1）提出了对于测量的图分割方法的要求，选择了适合的图分割方法对网络拓扑进行分割。并在此基础上提出了基于分区的链路传输重要度评价方法。利用得到的重要度筛选出了待测链路集并计算得到了对应的探测路径。

（2）提出了最小可识别集的概念，设计了对应的故障定位算法。对定位得到的可疑区域根据本文场景设计了全覆盖的探测路径选择算法。

（3）仿真并对比了全覆盖方法和本文方法的测量代价。考虑加入额外的链路信息对重要度评价进行优化，进一步减少了测量代价。

1.3 本论文的结构安排

本文的主要内容主要分为两部分：第一部分是对待测链路集的筛选，包括图分割、重要度评价方法和第一轮探测路径的选择，第一轮测量是周期性的测量；第二部分主要探讨了可疑区域的定位和分析，以及第二轮的探测路径选择，第二轮测量是非周期性的测量，是否进行二轮测量依赖于第一轮测量的结果，可疑区域是第二轮测量的主要目标。

本文分为六章进行阐述：

第一章是绪论，对网络测量的意义和重要性进行了介绍，并对本文的研究背景，思路进行了介绍。总结了本文的主要研究内容和贡献。

第二章是对本文研究内容的相关工作进行了介绍，包括网络测量，图分割算法，重要性评价算法，以及网络故障的探测和定位。

第三章首先阐述了本探测方法的主要设计思路。随后证明了相比于全覆盖的方法，基于分区的测量方法具有更低的代价，然后分析了本研究中的图分割要求，选择了适合的图分割方法。接着阐述了基于分区的链路传输重要度评价方法，并和对比方法进行了比较。最后针对选择得到的待测链路集提出了基于贪婪的探测路径选择算法。

第四章首先阐述了根据第一轮探测结果得到故障链路的故障定位方法，提出了最小可识别集的概念，和如何得到发生故障的可疑区域的方法。然后针对可疑区

域中未被一轮测量覆盖的链路，设计了基于全覆盖思想的探测路径选择方法。

第五章对探测代价和故障检出率进行了仿真实验。选择了典型的拓扑模型和常见的流量模式，在不同的待测路径比例下分别进行了仿真实验。评价对象主要是对一二两轮探测产生的总测量代价的平均测量代价和总的故障检出率。为了进一步减小测量代价，还考虑了额外的链路信息——链路利用率，并进行了进一步的仿真。

第六章是全文的内容总结和未来展望。

第二章 相关工作

本章介绍了一些相关技术。2.1 节分析了多路径测量的特点和重要性，介绍了两类测量技术——主动测量和被动测量，以及相关的研究工作；2.2 节介绍了节点和边的重要性研究，分别介绍了节点重要性和边重要性的主要指标和评价算法；2.3 节分别从图的局部划分和全局划分介绍了图分割技术，；2.4 节介绍了在以往的网络测量工作中的探针选择方法。

2.1 网络测量

为了提高网络资源的利用率，为客户提供优质服务，网络性能和状态应被迅速、准确地评估。网络服务质量和网络性能有着密切的关系，但是其侧重点不同。网络性能是指从网络运行的角度衡量网络实际运行的指标体系，而网络服务的质量则从用户的实际感受来描述服务的使用性能。

网络测量对于通信网络的正常运行是非常重要的，但是，测量通常会带来带宽、CPU 和内存使用方面的成本。网络测量旨在满足上述需求：它遵循某些方法，技术和标准，并利用某些测量工具和工具来获得运行状态和性能指标。网络测量是获取有关网络内部状态的有效方法,它为流量工程,网络组件管理和网络系统优化提供了可靠依据,并且为复杂网络理论研究提供了支持，带来了网络技术的发展。而为了有效且客观地评估网络性能,必须通过合理的性能评估方法对网络测量数据进行分析和处理,以帮助更好地了解网络状态。网络性能具有多种指标,不同的指标能够体现网络的不同状态。

随着计算机网络规模越来越大、结构越来越复杂，很多研究专注于大型网络中的主动测量方法。由于网络的异构性，被动测量在当今的计算机网络中是不切实际的，因为被动测量的前提是基于所有被管理实体都能发出故障告警。因此,基于端到端的主动测量探针技术成为解决网络测量问题的更好选择[58] 。

2.1.1 网络测量性能指标

为了全面，客观地评估网络性能，需要科学地选择指标以准确衡量网络性能。由 IETF 确定的 IP 网络性能指标包括：连接性，吞吐量，带宽利用率，时延，时延变化和丢包率等。下面简要介绍几个最常用的指标的具体含义：

1)吞吐量：指在给定的时间内的数据转发率。吞吐量衡量有多少数据包成功转发，通常以每秒位数（bit/s 或 bps）为单位，有时以每秒数据包（p/s 或 pps）或

每个时隙的数据包为单位。

2)丢包率：分为端到端的丢包率和端口之间的丢包率。端到端的丢包率可以通过源端发送的数据量减去目的端接受的数据量再除以发送数据量得到；端口之间的丢包率指的是链路上端口间的丢包率，与端到端丢包率的计算方法类似。

3)时延：分为单向时延 (OWD) 和往返时延（RTT）。单向时延，又称端到端时延，指数据包从指在从源到目的端之间经由网络传输所用的时间；往返时延是发送数据包所用的时间加上接收数据包的确认所用的时间，其中包括两个通信端点之间路径的传播时间。

4)带宽利用率：带宽利用率是链路上已用的带宽占链路上可以使用的总带宽的百分比。能够反映一条链路的拥塞和负载情况

无论是在传统网络还是在 SDN 网络中，时延、带宽、丢包率和时延抖动都是网络测量最重要和最常见的性能指标。本文所述的方案针对端到端多路径的测量，可以以这些性能指标作为路径的状态判断依据。当测得路径指标大于给定阈值时，判断路径发生故障，触发故障定位过程。

2.1.2 主动测量技术

主动测量技术是指向网络中发送探测包，根据探测包得到的信息对网络状态进行推测和判断。最典型和常见的主动测量工具如 Ping、Traceroute 等。Ping 可以判断网络的连通状态，获得往返时延，丢包率等信息。Traceroute 可以得到端到端的路由信息。由于主动测量无需运营商配合，更为灵活，因此广泛被用于网络状态测量中。

主动诊断技术可以自适应地选择探针，并且不需要多余的仪器。根据所选探针的结果进行诊断。主动测量不依赖于测量对象本身的测量能力，并且适用于底层端到端网络性能的测量。缺点是发送的探针不仅占用网络资源，而且可能会影响网络本身的运行。

随着 SDN 的出现，通过控制器就能够获得底层转发设备全局拓扑，还能够获得相关的性能状态数据。研究人员对适应 SDN 和 OVS 的测量方法越来越感兴趣。一些系统使用镜像进行测量。NetSight[84]使用镜像来收集网络中所有数据包的轨迹信息。然而，他们的方法只适用于小的网络，不具有可扩展性。Everflow[85]在数据中心网络中提供了一个可扩展的数据包采样。然而，Everflow 需要特定的硬件，例如多路复用器，来支持它的采样方法。此外，他们的方法是反应性的，用于对事件的响应，[86][87]研究了在基于 openflow 的网络中使用探测包来测量延迟。SLAM[86]使用 OpenFlow 报文到达控制器的时间来估计链路之间的延迟。然而，这种方法需

要了解流量模式，所以它可能不适用于非数据中心的网络。OpenNetMon[87]系统为OpenFlow 网络的每个数据流提供了吞吐量、延迟和丢包率等指标。它使用探测包和一个收集器来进行延迟测量。Pingmesh[88]使用 ping 来测量数据中心的延迟。但 OpenNetMon 和 Pingmesh 不提供自适应测量框架。SDProber[77]是一种主动的，自适应的延迟测量方法，SDProber 调整了转发规则，使探测报文更频繁地路由到容易发生拥塞的区域。能够及早发现高延迟。

2.1.3 被动测量技术

被动测量是非侵入性的，在关键位置和节点设置测量设备，通过捕捉和统计分析数据包，可以获得网络状态和性能的参数，被动测量不发送任何其他探测数据包，因此不会影响网络行为，测量结果将更加准确。

被动监视通过窥探现有的网络流量来推断网络性能。执行被动监视有两种方法[82]:

●两点监视:这种监视方法在每个被监视流量的入口和出口节点部署两个监视设备。性能指标是通过比较在入口和出口监视器上执行的度量来推断的。这需要同步监视器的时间戳，并识别遍历它们的所有包。然而，当穿越监视器的通信量很重要时，识别过程可能会导致严重的可扩展性问题。

●单点监视:这种监控方法需要一个监视器来监控一个流程。它使用 TCP 确认来推断部署监视器的点和被监视的 TCP 流的接收之间的性能度量(例如，丢失率和在监视器位置和被监视的流的接收之间的段上的往返时间)。这种方法的应用仅限于 TCP 流。

被动测量可以在数据包级别或流量级别进行。数据包级别的测量基于精细的粒度，包括源和目标地址、端口号、数据包大小、协议和特定的应用程序层数据。流量水平测量需要适当的聚合规则，并且收集的数据包括流量、流量大小和流量分布。为了减少测量数据，非常有必要使用采样技术来减少测量开销。但是，相比于主动测量，被动测量更为复杂，并且只能获取部分网络数据。其准确性取决于测量设备的性能以及所采用的统计和分析算法。

2.2 节点和边的重要性

近年来，对复杂网络的研究成为许多不同学科的研究热点。几乎所有的复杂系统(比如社交关系，食物链，道路交通系统)都可以表示为网络。其中, 节点代表系统的各种构成要素, 节点间的连边表示要素之间的联系[9]。节点和边的重要性研究是指基于节点或边重要性排序算法，对复杂网络中的节点或边进行排序，更好的来

了解网络，从而对网络进行管理和优化。

2.2.1 节点重要性

如何识别重要节点在复杂网络分析中是一个很重要的问题，所谓重要的节点是指与网络的其他节点相比，对网络结构和功能有较大影响力的节点。重要节点的影响能够迅速扩散到网络中的大部分节点，但其数量一般很少[9]。

在不同的场景中，节点重要性具有不同的评价角度，但一般都是根据网络的结构特征来构建节点重要性的评价算法。主要可以分为四类：基于节点近邻的方法、基于路径的方法、基于节点移除和收缩的方法和基于特征向量的方法[11]。

基于节点近邻的方法是最简单、最直观的方法，根据节点的直接邻居数目来评价节点重要性。最典型的是度中心性(degree centrality)算法[5]。度是最简单的特征，也是最早被运用于识别网络重要节点的特征。度中心性算法具有简单直接的特点，虽然需要了解整个网络拓扑信息，但计算的复杂度很低。由于只考虑了局部信息，基于度中心性选择的重要节点在很多情况下可能是不精确的。为了弥补这一不足，后续工作中提出了很多改进算法，如 Chen 等人[15]提出了一种基于半局部信息的节点重要性排序方法，引入了全局信息；此外，Kitsak 等人[14]网络中的位置信息引入节点重要性评价，提出用 k-shell 分解法(k-shell decomposition) 将外围的节点层层分离, 处于内层的节点具有较大影响力。

在交通、通信、社交等网络中节点的重要性常常体现在其传输信息的能力，一些节点可能具有很小的度但却是连接几个区域的“桥节点”, 它们在流的传递中担任重要的角色。此时, 基于近邻关系的算法无法很好的衡量节点重要性，需要考虑网络中节点对信息流的控制能力, 而信息流的控制能力和网络中的路径有密切的关联[9]。基于路径的常见方法有接近中心性(closeness centrality) [5]和介数中心性(betweenness centrality) [6]等。接近中心性是其中典型的算法，该算法认为与网络中其他节点最接近的（平均路径最短的）节点是重要节点。介数中心性也是基于路径特征设计的算法，在介数中心性中，节点重要度和节点在其他节点组成的节点对之间的最短路径上出现的频率相关。具有较大的介数值的节点两端可能连接不同集群，对复杂网络的研究具有重要意义。

基于节点移除和收缩的方法的核心思想是破坏性等于重要性[11]。当移除节点对网络造成的破坏性越大，节点越重要。破坏性体现在多个方面，除了直观的连通性，还可能影响到网络的其他结构特征如最短路径[12]，生成树数量[14]等。但这类方法计算复杂度很高，目前相关的实验仍局限于小型网络。

基于特征向量的方法评估了节点的质量对节点重要性的影响。PageRank[8]是

这类方法中最著名的算法，它用于评价 web 节点的重要性程度，是 Google 搜索引擎的核心算法。PageRank 主要是基于链接页面的数量和质量来评价该网页的重要性。PageRank 及其改进算法（如 LeaderRank）广泛应用于其他领域如对期刊的排序、 对社交网络上用户的排序等[9]。考虑到节点质量和枢纽性，研究人员提出了 HITs 指标[16]、自动资源汇聚算法[17]、SALSA[18]这三种算法来处理更复杂的节点重要性排序。

2.2.2 边重要性

相对于节点重要性研究的丰富多彩，对边重要性的研究相对较少。最早关于边重要性的研究由 Mark Granovetter 在 1937 年提出，他指出弱连边的重要性可能比强连边更重要[19]；Cheng 等人[10]考虑了文档网络上的一个类似的问题，其中文档之间纽带的强度是以内容相似度为特征的。他们发现连接较少相似文件的纽带在维持整体联系方面更为重要；Onnela 等人[22]认为网络中两个节点的共同邻居节点越多，它们之间边的连接强度越强；Radicchi 等人[20]将节点的聚集系数推广到边，认为聚集系数较低的边一般处于社区之间；Gilbert 和 Karahalios 等[21]基于用户特点和交互行为考虑两个节点的属性信息和交互强度。

由于网络中边的数目多于节点，且边的主要功能是作为网络中流的载体，其重要性的度量和排序比节点难度大很多，但在研究人员的努力下依然取得了一些成果。Girvan 和 Newman[23]基于介数中心性提出了边介数，即网络最短路径中经过该边的比例，边介数越大说明该条边在网络中的位置越重要，但边介数需要计算每对节点的最短路径数，计算耗费时间长；De 等人[24]基于随机游走提出了的k-路径介数，在网络中信息经过k次传播，经过次数越多的边重要性越高：姜等人[25]提出了基于网络传输特性的链路重要度评价方法，其基于链路在最短路径中的使用频率来评估链路的重要性，使用频率最高的链路最重要。

综合来看，目前对边的重要性研究还相对欠缺，已有的边重要性排序算法存在一些不足，例如计算复杂度高，没有综合考虑边的位置信息和传播信息的能力，算法准确性有待提高。

2.3 图分割算法

当对应用程序问题进行建模时，计算机科学家经常使用图作为抽象。将图切成更小的块是基本算法操作之一。即使最终的应用程序涉及不同的问题(例如遍历、寻路、树和流等)，对大型图进行分区对于降低复杂度或并行化来说通常也是一个重要的子问题。图分区（Graph Partitioning）因此也变得越来越重要，图分区问题

在很多方面有重要应用且具有挑战性。图分区是经典的 NP 完全问题，很难找到其最优解。具有精确解的分区只有在具有有限顶点且边非常少的图才可能存在。研究人员通过不懈努力，提出了很多具有较好性能的图分割算法，主要有谱划分方法，几何方法，启发式方法，多路划分方法等。

2.3.1 图分割概念

理解图分割方案及其目标函数需要以下概念。在图$G = (N, E)$中，如果存在边$u, v \in E$，则节点v就是节点u的邻居。B_i是图划分得到的块，如果节点$x \in B_i$有一个邻居节点$y \in B_j$，$i \neq j$，则x和y称为边界节点,块与块之间的边,如（x，y）称为割边。设置$E_{ij} = \{\{u, v\} \in E: u \in B_i, v \in B_j\}$是两个块$B_i$和$B_j$（$i \neq j$）之间的割边。度$d(v)$是节点$v$的邻居数。图的邻接矩阵$A$是描述节点连通性的$|N| * |N|$矩阵。矩阵的元素$A(u, v)$指定节点$u$到节点$v$的边的权值。如果节点之间没有边，则权值设为 0。图 G 的拉普拉斯矩阵定义为$L = D - A$，其中D是表示节点度的对角矩阵，A是邻接矩阵

图分割算法最突出的目标函数是最小化总切，即：

$$\sum_{i<j} w(E_{ij}) \tag{2-1}$$

其中$w(E_{ij})$是E_{ij}中所有边的权值之和。

其他的图分割形式如通信量，对于块B_i通信量定义为$comm(B_i) = \sum_{b \in B_i} c(b)D(b)$,其中$D(b)$表示 b 具有邻居节点不同块的数量，不包括$B_i$。然后，最大通信量定义为$max_i$ comm（B_i），而总通信量定义为$\sum_i commB_i$。尽管使用其他的分区目标例如通信量或块形状（由块的长宽比形式化），有利于某些应用场景，但在图分割领域已将最小化总切作为一种标准。原因之一是切割优化在实践中更容易；第二是对于具有较高结构局部性的图，切割通常与大多数其他公式相关。

图的均衡分割，即将图划分为 k 个大小大致相等的块，以使割边度量最小化，这是一个 NP 完全问题[30]。

2.3.2 图分割相关研究

常见的图分割方案可以分为全局方法和局部方法两种。局部搜索是一种简单而广泛使用的优化元启发式方法，它通过从邻域中选择一个新的解决方案来迭代地改变一个解决方案。全局方法是使用处理整个图的方法，直接计算一个解决方案。这些算法通常用于较小的图形，或作为子程序应用于更复杂的方法，如局部搜索或多层算法。关于最优求解图分割问题的方法有大量的文献，这些方法中有许多局限于二部划分，但通过递归可以推广到 k 级划分。

2.3.2.1 局部方法

最早且著名的局部方法是 Kernighan-Lin 算法[4]，在这个算法中，一个选择策略被用来寻找节点分配的交换，使总切割大小减少最大。当所有节点对被交换后，迭代结束，分区被重置为迭代中遇到的最佳节点交换，当迭代不能找到改进时，算法终止。KL 方法的一个主要缺点是，它的时间代价非常高。KL 算法的每次迭代需要$O(|E|log|E|)$时间。Fiduccia 和 Mattheyses[28]改进了 KL 算法，使用了单点移动和桶列表，降低了时间复杂度。Karypis 和 Kumar[29]进一步加速了算法，只允许交换边界节点，并在移动节点后，没有减少割边数时停止迭代。Holtgrewe 等人每次移动一个节点，允许在减少割边或改善平衡之间进行更灵活的交易[32]。Diekmann 等人[33][34]在二分划情况下引入了更普通的邻域关系，整个集合的节点在块之间交换，而不是迁移单个节点，以提高减少割边数。该算法的运行时间与 KL/FM 算法相当,而分区的质量往往优于其他方法[34]。

2.3.2.2 全局方法

图分割的全局方法依赖于图的性质，而不是任意的初始划分。全局法最常见的例子是谱划分，它利用邻接矩阵的谱来进行划分。光谱划分技术最早由 Donath 和 Hoffman[35][36]以及 Fiedler[37]使用。谱二分法通过计算图的拉普拉斯矩阵L的第二最小特征值所对应的特征向量来推断图的连通性的全局信息。对应的特征向量也被叫做 Fiedler 向量，它根据对应向量项的符号将图分成两个团体，也是切割优化的解。分割的方法是确定特征向量中的中值m，并将入口小于或等于m的所有节点分配给B_1，其余节点分配给B_2。为了获得 Fiedler 向量的快速近似，Barnard 和 Simon [38]使用了一种多层方法，该方法通过独立的节点集进行粗化，并通过 Rayleigh 商迭代进行局部改进。Hendrickson 和 Leland[39]将谱方法扩展到利用多个特征向量将图划分为两个以上的块，并将该方法推广到具有节点权值和边权值的图上。

获得图的二部划分的一种简单方法是图增长[40][41]。它的大多数变体都是基于广度优先搜索(BFS)的。最简单的版本如下所示。从一个随机节点v开始，使用从 v 开始的 BFS 将节点分配到节点集$V1$。在将原有节点权值的一半赋给这块后，搜索停止，并将$V2$设置为$V\backslash V1$。该方法可以结合局部搜索算法来改进分区。也可以通过观察一个与随机种子节点距离最大的节点[41]来寻找一个好的起始节点。该算法的变化总是将导致切增量最小的节点加入到块中[40]。

利用著名的最大流最小割定理[42] ，可以通过计算最大流量和它们之间的最小割来划分图中的两个节点集。但这种方法没有考虑图的均衡分割，如何将其应用于均衡分区的图分割问题是一个挑战。

在某些情况下，分区可以利用图节点在空间中的坐标，这样的方法称为几何方法。使用节点坐标进行划分有很多好处，比如递归坐标二分法(RCB)[43]和递归惯性分割法[44][45]。RCB 将图节点投影到平面坐标最长的一个维度上，并通过它们投影的中位数将它们平分，等分平面与坐标轴正交。RCB 是一种递归二分法，执行速度很快，但该算法得到的分区质量低，子区域可能不连通。递归惯性分割作为 RCB 的一种改进算法，降低了不连通子区域的数量。递归惯性分割的平分平面正交于一个使到所有节点的距离平方和最小的平面L。Miller 等人的随机球算法[46][47]对 RCB 算法进行了推广，将d维节点立体投影到一个$d+1$维的随机球上，该球通过其中心点被一个平面等分。基于几何的分割算法的其他代表还有空间填充曲线法[48][49]，空间填充曲线是一种基于多维降维的分割方法，为了保持节点在空间中的局部性，定义了从V到$\{1,\dots,|V|\}$的双射映射。采用的双射映射使得空间填充曲线法比 RCB 更简单、代价更小。图形填充曲线[50]的方法将空间填充曲线推广到一般图形。其他的工作试图通过使用多级图绘制算法[51]将任意图嵌入坐标空间，将图结构的信息带入几何图形中。

2.3.2.3 多层方法

上述方法具有各自的优缺点，都能得到较好的解。但为了结合各自的优点，提高算法的整体性能，并且能够得到对规模较大的图的较好分割，研究人员结合上述方法提出了多层划分方法。

为了改进 k 分区的局部搜索算法，对 KL/FM 算法提出了多种扩展。一种早期的 k 路局部搜索算法使用了$k(k-1)$个优先队列，每个优先队列对应一种移动类型(源块，目标块)[53]。在单次的移动中，选择使增益最大化的节点，通过这样均衡的移动使分区逐渐优化。Karypis 和 Kumar [52]提出了以线性时间运行的 KL/FM 算法的k路方法。他们为所有类型的节点移动使用一个全局优先队列。使用的优先级是最大的局部增益，即当节点被移动到它的一个相邻块时，对割边的数量有最大的减少。所选择的节点对目标产生最大的改进，并且满足图分割的均衡约束。目前大多数的局部搜索算法在分区块之间交换节点，试图减少割边数，同时保持分区平衡。这极大地限制了改进的可能性。Sanders 和 Schulz[54]放松了节点移动的平衡约束，通过结合多个局部搜索来保持或改善全局的平衡。他们将组合问题简化为消除图中的负循环，并利用了现有算法来解决问题。

Sanders 和 Schulz [55]引入了一种基于最大流最小割的技术，以改进给定二分法的边割。该算法通过在给定的边界节点/割边周围增长面积来构造一个 s-t 流动问题。区域中的每个 s-t 割对应于原图的一个充分满足平衡约束的二分解。然后采用

最大流最小割算法来获得该区域的最小切割。后来的工作对该方法进行了多种改进，如迭代地应用该方法，在更大范围内搜索可行的分割，或通过使用给定的最大输出量，应用启发式方法输出更均衡的最小分割。

Diekmann 等人[56]扩展了图增长和之前的工作[57]，得到了一个称为 Bubble framework 的迭代过程，该迭代过程能够划分$k > 2$个井字形的块。该方法的应用受益于原图良好的几何块形状。首先通过仔细选择均匀分布在图上的k个种子节点来扩展图的增长。从k个种子节点开始，第二步将使用k个广度优先搜索，利用图增长算法来分割块，注意，只能让最小的块接收下一个节点。在这一步骤中进一步使用局部搜索算法来平衡分割块的负载并改进分区的结果，但可能产生未连接的块。迭代的最后一步为下一轮计算新的种子节点。块的新中心被设计为使到块内所有其他节点的距离之和最小的节点。算法的第二步和第三步将迭代 10 次以上，直到种子节点停止变化或没有发现改进的分区。

聚类也被认为是一种节点划分方案，在聚类方法中，平衡分区的约束被移除。图的分割也被认为是图节点的聚类，但是它们都有不同的目标来最大化或最小化特定的目标函数。

2.4 故障检测与定位

故障在网络中是不可避免的，如何即时的处理故障对于网络运行的可靠性是非常重要的。网络中对故障的处理一般分三步：第一步是确定网络中是否有故障发生；第二步，对故障发生的位置进行确定；最后，对定位到的故障进行恢复。在这三个步骤中，故障的检测和定位都和网络的状态测量有着密不可分的关系。在网络的故障检测中，通常没有直接的办法来获取网络中的具体故障，一般是通过对测量得到的网络状态来判断网络是否发生故障和发生故障的具体位置。

最近的网络监测工作将故障定位作为故障检测的一种反应，并进行两阶段监测[78]。在检测阶段，希望能使用尽可能少的网络资源来检测网络中是否存在故障。并在检测到故障时，返回一组可疑链接。然后是定位阶段，目的是将这组可疑链路范围缩小到故障链路。根据网络测量的方法，对路径的监视可以采用主动方式或被动方式。在网络层析成像领域也有一些相关工作，用于辅助分析网络中的故障检测和故障定位。但在实际应用中，很多网络设备都没有这样的功能，在传输过程中也可能会出现告警中断或丢失的情况。同时，现代网络的复杂性使得故障报警与真实故障之间的关系复杂化，难以正确识别故障位置。近年来，相比于被动方法，主动测量方法由于几乎不受网络组件的支持的限制，并且能够快速准确地推断网络性能，得到了广泛的关注。

主动测量方式需要网络注入称为探针的数据包，探针是一个实体，可以执行一个或多个测量操作，并获得网络数据和性能参数。在故障处理过程中，根据功能的不同，在探测阶段使用的探针，称之为故障检测探针，在故障定位阶段使用的探针，称为故障定位探针。网络测量探针可以收集各种不同网络，尤其是大规模网络的数据或数据包的踪迹。探针的设计和选择是故障处理中最核心的部分测量与评估系统根据数据流可分为三个层次：数据采集，数据分析以及数据表示。通过各种测量探针收集了网络性能数据和流量数据。性能和行为分析取决于特定的数据分析过程和数据库软件。探针本身是一个数据收集器，可以独立实现 ICMP_Ping，UDP_Ping 和 Traceroute 等功能。

2.4.1 故障检测探针

为了发现网络中存在的故障，需要向网络中发送用于故障检测的探针组，以获取相关的网络内部状态信息。探针的发送和接收产生操作成本，并导致额外的网络负载。因此，执行故障检测任务的探针组应该适当地选择。具有限制成本的最具信息量的探针组是最佳的。但是，选择此最佳探针组是 NP 难问题。探针的选择分为预先计划的策略和适应性的策略，目前最广泛被使用的方法是一种启发式贪婪算法，该算法反复依次选择最有用的探针。每当要选择的探针，每个探针的质量（其通常是由所述探针的相互信息量化）被计算。将选择相互信息量最大的探针[58]。由于相互信息的计算复杂度非常高，因此探针选择过程会占用主动诊断的大部分时间。

Rish 等人[61][62]和 Brodie 等人[60[60]提出了一些成本效益高的自适应诊断探测技术。这些技术利用一种信息理论方法来选择目标探测集，首先选择一个信息最丰富的小探测集，然后根据网络的观测状态动态调整目标探测集。郑等人在[63]中提出了另一种基于熵近似的方法。该方法采用环路信念传播模型计算边缘熵和条件熵的近似值。

Natu 等[65-72]研究了选择目标探针集的适应性策略。在这些自适应策略中，将故障诊断的探针选择步骤分为故障检测和故障定位两个步骤。因此，当检测到故障时，故障检测步骤触发故障定位步骤。在故障定位步骤中，根据探测结果确定可疑区域，并发送附加探测来确定故障在网络中的确切位置。Lu 等人[70]提出了另一种基于[66]中思想的自适应方法。该方法将故障检测过程划分为一系列阶段，在每个阶段中选择一个小的探测集对几个网络节点进行检测，直到覆盖网络中的所有节点。Ayush D 等人[76]提出探测的成本由探测的长度决定，探测的长度是它所遍历的链路的数量。考虑使用网络分区来生成区域内的候选探针集，该候选探针集不增加

探针选择算法选择的目标探针集的开销。

2.4.2 故障定位探针

由于故障检测探针所获得的网络状态信息并不足以将检测到的故障定位到具体的故障源，而是返回一个包含故障的可疑范围，因此需要向网络中发送附加的探针以取得额外的状态信息，从而将可疑范围缩小，最终定位到具体的故障源。这样的附加探针被称为故障定位探针。和故障检测探针不同，故障定位探针并不确定，也不按周期发送，而是根据探测探针返回的不同结果进行“反应式”的选择和发送。

Brodie 等人在[59]中首次提出了探测诊断网络故障的应用，解决了单节点的定位问题，描述了几种选择网络目标探测集的近似算法。Tang 等人在[64]中提出了一种新的故障定位技术——主动集成故障推理。在这种技术中，将网络症状的相关性与主动探测结合使用，以选择用于故障定位的目标探测集。根据网络中观察到的症状调整目标探针集。所有增量自适应探测方法都非常适合于实时监测和诊断，因为探测是根据需要选择和发送的，以响应发生的故障。Carmo 等人[73]提出了在无线多跳网络中使用主动探测的入侵检测方案。提出了一种递归探针选择方案，用于选择目标探针，并将探针的结果输入贝叶斯分类器来推断网络节点的状态。Garshasbi 提出了[74]另一种结合主动和被动监测方法进行故障诊断和定位的算法。该方法是基于蚁群优化算法[75]中的一种，该算法通过将计算问题简化为在图中选择有效路径来解决计算问题。S. Pan[80]将拥塞链路识别过程建模为马尔可夫决策过程(MDP)，然后采用一种强化学习技术，即Q-Learning 来求解该 MDP。

网络层析成像，又称网络断层扫描，是一种用于监控网络中链路性能的方法。它还包括从相关的端到端测量来推断诸如链路丢失和包延迟等内部性能，通过描述从起点到目的地(OD)流流量的流时间，以监控各种 QoS 故障[31]。由于单个组织只能直接访问网络内部节点的有限部分，而且由于可能存在商业方面的冲突，它们很难在共享内部性能观察数据方面进行协作，为此提出了布尔层析成像。网络布尔层析成像(Boolean tomography，BNT)提供了通过端到端监控路径来评估网络状态的工具，因为它们不依赖于管理访问权限。布尔网络断层扫描技术克服了传统的基于广泛部署的监控代理(如 SNMP)或广泛支持的网络协议(如 traceroute)的网络监控方法由于现代计算机通信网络的复杂性和异构性而面临的局限性。事实上，各种客户软件和网络功能中的错误和配置错误会导致“静默故障”，这只能从端到端连接状态检测到[82]。网络布尔层析成像可以用来定位和识别拥塞链路。它用布尔代数解决了端到端路径的状态测量与内部链接的拥塞状态相关的方程组。网络断层扫描阶段的缺点是计算难度大，且推断准确率较低。

2.5 本章小结

本章对网络测量技术，节点和边的重要性评价方法，图分割算法及其相关研究，以及网络故障检测的测量和定位的探针选择进行了介绍。这些技术都为本文的研究提供了启发和指导，本文的测量方法正是基于以上的技术和相关研究提出的。

第三章 基于分区传输重要度排序的待测链路选择

3.1 设计思路

网络中的路径状态包括，延迟、带宽、丢包率等。当一个链路发生故障时，它的性能会显著下降(例如，网络丢包和长延迟)。主动测量提供了关于拥塞和其他网络故障的有价值的信息。在故障检测中重要的是尽早主动发现长时间存在的故障，并在它开始时尽快处理。可以通过定期的主动测量来判断网络链路的性能[77]。

对于网络中的故障检测，根据测量对象的不同，可以分为针对节点故障的检测（如节点失效）和链路故障的检测（如链路拥塞）。在一般情况下，检测所有链路级故障的一个充分必要条件是要覆盖网络的所有链路。本文将这种类方法称为全覆盖的方法。如果一条链接被至少一条探测路径遍历，则称该链接被覆盖。则全覆盖方法要求需要监视一组覆盖所有网络链路的路径，这些路径的并集等于网络链路集。如果一条链接被至少一条探测路径遍历，则称该链接被覆盖。链路全覆盖是最常见的检测链路级故障的方法，最典型的方法有贪心和减法[81]的方法，这些方法的主要思路都是产生相应的探针实现对网络中测量对象的全覆盖，从而得到整个网络的故障发生情况，全覆盖方法的主要实现流程如图 3-1。这样的探测并不能确定故障的发生点，但能够检测出网络中是否存在故障。全覆盖的思路最大限度的确定了网络中故障的存在与否，但全覆盖的方案会带来大量的额外流量，测量的代价很高。

在常见的故障检测过程中，探针站通过发送预先选定的一组探针来周期性地对网络进行探测。探针结果被分析以检测故障或性能问题的存在。由于这些探测即使在网络健康时也会定期运行，因此应该将探测集最小化，以施加最小的网络探测流量。本文希望找到一种不需要覆盖全部链路，但仍能保证较高的故障检测率的检测方法。

通过对拓扑进行观察，可以发现在某些规则拓扑中，比如数据中心常见的胖树拓扑，如图 3-2（a），胖树拓扑通常是一个绝对对称的拓扑，它的边缘交换机和汇聚交换机组成一个个 pod。当网络中流量模式为随机均匀的，每条链路都可能发生

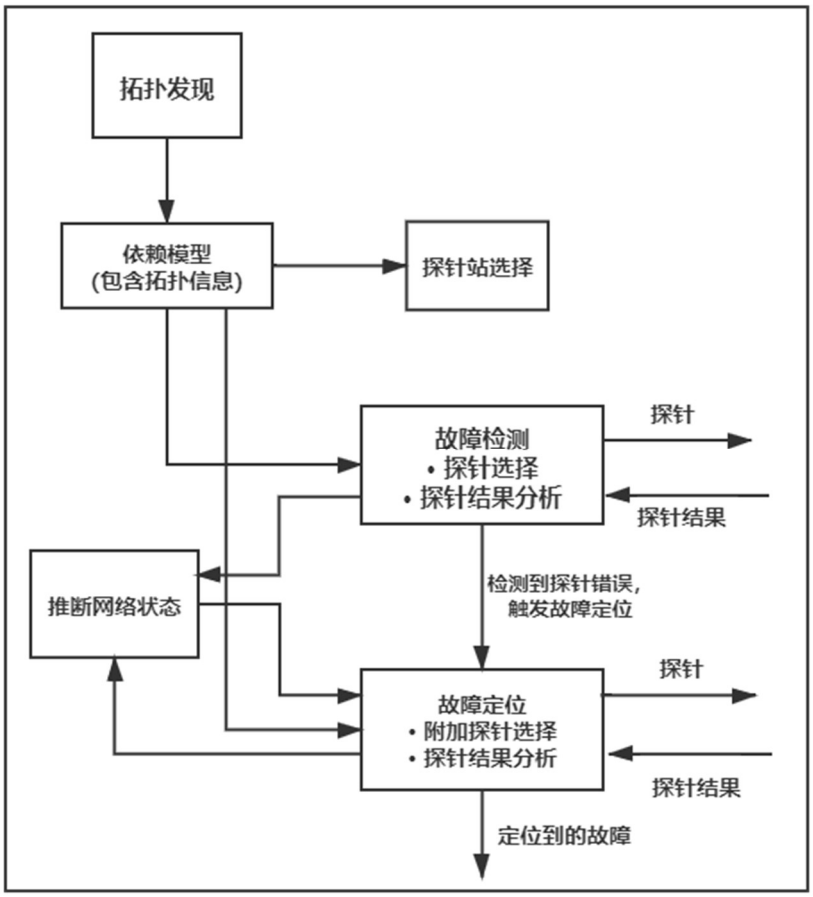
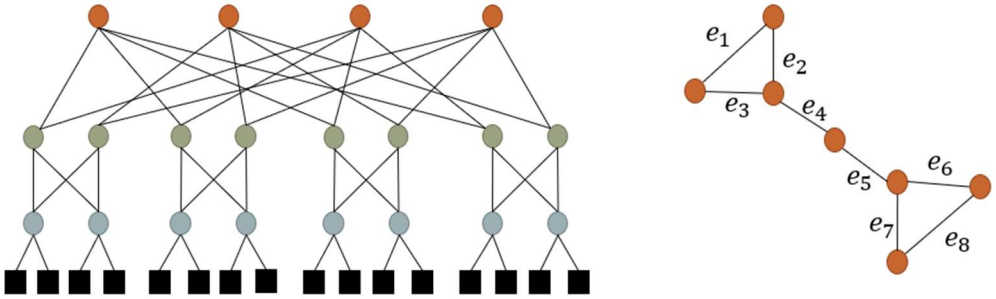


图 3-1 全覆盖测量方法主要测量流程

故障，并且位于相同的层之间的链路都具有相同的拥塞可能性，它们的重要性几乎没有差别。但对于不规则拓扑，如图 3-1（b）情况会有所不同。在所有流量都通过最短路径传输的前提下，某些链路，如e_1、e_8，很难通过两条以上的流，发生拥塞的可能性很小。基于此，为了减小主动测量方法所引入的外部流量，研究者们提出根据链路先验拥塞概率的探针选择，如[79]，但学习链路故障的先验概率本身就会引入额外流量，并且在网络发生变化的时候，先验知识可能会失效，从而带来新的问题。本文所采用的探测路径选择方法只需要知道网络的拓扑结构，无需获得和故障概率相关的先验知识。

本章通过对拓扑进行分区和对链路的传输特性进行分析，利用链路的传输重要性，选取“最重要”的部分链路作为反映区域网络状态的待测链路。由于未覆盖所有链路，漏检是不可避免的。为了对未覆盖的链路进行探测，本文选择对网络拓扑进行分割，将所选的“最重要”的链路的状态作为其所在区域的状态。

本章所提出的检测方法和全覆盖方法的主要区别在于探针选择方面。本章将从拓扑图分割，链路重要性评估，和基于待测链路集的探针选择三个方面进行介绍，并在 3.5 中设计和实现了对上述内容的仿真。



（a）　　（b）

图 3-2 规则拓扑和不规则拓扑示例。（a）胖树网络：同层链路重要度一致，整体区别不大；（b）不规则网络：链路重要性差别大

3.2 拓扑图分割

将一个大问题划分为子问题在为各种各样的问题开发有效的解决方案中起着关键作用。通过将任务划分为子任务并单独处理每个子任务，可以降低与任务大规模处理相关的复杂性。一个好的分区方案可以极大地减少解决复杂问题所需的内存和时间。在本文中，将网络分区有助于简化网络拓扑进而减小故障检测开销，这在 3.2.1 节进行了详细证明。同时在利用检测结果在推断故障点的过程中，探测产生的开销和定位故障点的复杂程度与网络拓扑的规模有着密切的关系。通过对网络拓扑进行分割得到小规模的区域，对得到的可疑区域进行进一步的探测，可以显著降低故障定位的开销，进而减小定位探测代价和计算量。

3.2.1 分区方案和全覆盖方案的对比

设有网络拓扑$G = (N, E)$，其中 N 表示网络中的节点，E 表示网络中的链路，设每条边权值（w_{e_i}）表示在一个时间间隔中链路i发生故障的概率。选取网络中最易发生故障（即权值最大）的m条链路作为待测路径集$\{e_1, \dots, e_m\}$。

对网络拓扑的分区可以看作是一个图分割问题。将网络划分为k个区域（$k \geq 2$），为$Z = \{Z_1, Z_2 \dots, Z_k\}$，即：

$$Z_1 \cup \dots \cup Z_k = N \tag{3-1}$$

$$Z_i \cap Z_j = \emptyset, \ \forall i \neq j \tag{3-2}$$

每个区域中的链路数为$\{N_1, N_2 \dots, N_k\}$，通过测量待测链路探测到每个区域发

生故障的概率为：

$$P_i = 1 - \prod_{j=1}^{N_i} \left(1 - \mathrm{w}_{e_{ij}}\right),\ if\ e_{ij} \in \{e_i, \ldots, e_m\} \tag{3-3}$$

其中e_{ij}为区域Z_i中的链路，当探测到区域存在故障时，将对存在故障的区域中的链路进行进一步测量，以确定是否有其他故障存在。对于一次测量，令测量代价为C，将探测一条链路一次的代价定为 1。假设判断一个区域z_i的链路状态的代价为C_{z_i}，由于很难构建出不重复的测量路径，显然$C_{z_i} \geq N_i$，则对于一个时间间隔，令分区方案总的测量代价为C，$E(C)$为C的期望：

$$E(C) \geq m + \sum_{i=1}^{k} P_i * C_i \tag{3-4}$$

而对于全覆盖测量方案来说，测量代价C_{all}为:

$$C_{all} \geq |E| \tag{3-5}$$

因为$\forall\ Z_i \in Z, P_i \leq 1$，显然有：

$$C_{all} \geq C \tag{3-6}$$

对于式(3-6)，当且仅当在同一时间间隔所有区域的待测链路都发生故障，等号才成立。事实上，网络中的故障总是稀疏的，所有区域同时发生故障的概率非常低，分区检测的代价总是会小于全覆盖的代价。

相对于全覆盖的方案，由于基于分区的测量方案在故障检测中没有检测所有的链路，对未覆盖链路造成的故障漏检是不可避免的。区域的划分和待测路径集的选择都会对故障检测准确度造成影响。本文对拓扑图使用的图分割方法将在 3.2.2 中进行讨论。

3.2.2 图分割

3.2.2.1 图分割基础理论

通常意义上的图分割可以分成两类：基于点的划分和基于边的划分。两者的主要区别在于，基于边的划分中产生割边，割边会被复制到不同的分区中，而基于点的划分中产生割点，割点会被复制到不同的分区中。图 3-3 是两类分割方式简单示例，虚线代表分割方式。

在实际的应用中，这两类划分方法各有优劣。比如在常见的分布式计算中，采用基于边的划分可以节省存储空间，但会带来额外的通信开销，而基于点的划分可以减小通信开销，但会增加存储开销。在常见的链路状态路由协议中，OSPF（开放最短路径优先协议）的区域边界在路由器上，路由器的不同接口分属不同区域，

可以看作是基于点的划分；而近年来被广泛使用的 ISIS（中间系统到中间系统）协议中，每个路由器都属于只属于同一个区域，边界位于路由器之间的链路上，可以看作是基于边的划分。在本文中，由于本文针对链路级故障的测量方法设计，所以主要研究的分割方式是基于边的划分。

（a）　　（b）

图 3-3 基于点的划分和基于边的划分的区别。（a）基于点的划分；（b）基于边的划分

割边是基于边的图划分问题中不可忽视的问题，讨论图的分割首先需要介绍割边。本文中所用到的图是基于网络拓扑抽象得来的，因此在本文中将图分割产生的割边称为域间链路。

定义 1（域间链路）：对于拓扑图给定的两个分区Z_i, Z_j，其中$i \neq j$。若Z_i和Z_j中任意两点间存在链路，则称这样的链路为域间链路。

令s_i为Z_i的负载，在本文中表示为Z_i中节点的数量，设平衡因子τ表示不同分区之间的不均衡程度，$cross-edge$表示将拓扑图进行分割产生的总域间链路数。$cedge(Z_i, Z_j)$表示Z_i, Z_j之间的域间链路数，将上述元素用数学符号表示：

$$1-\tau \leq \frac{s_i}{s} \leq 1+\tau, i \leq 1,2 \ldots, k \tag{3-7}$$

$$cross-edge = \sum_{i=1,j=1}^{n} cedge(Z_i, Z_j) \tag{3-8}$$

3.2.2.2 图分割方法

本文的目的是为了用区域中的部分链路来表示整个区域的故障情况，这对分区算法提出了三个要求：

(1)划分得到的区域是连通的。如果划分得到的区域是不连通的，区域内的进一步检测会没有可以检测的链路；从另一个角度来说，如果区域内不连通，那区域内节点和区域外节点必存在链路，也就是说域间链路可能会更多。总而言之，如果划分得到的区域内不连通，那么有更大的漏检的可能。

(2)划分尽量均衡。均衡的分割是让每个区域具有尽量相同的节点数和链路数。当划分不均的时候，具有较多链路和节点的区域发生故障的可能性会更大，大的区域会带来更多的区域检测代价，总体检测的代价可能会增加，具体证明如下:

假设将网络划分为k个区域（$k \geq 2$）,为$Z = \{Z_1, Z_2 \ldots, Z_k\}$，每个区域中的链路数为$\{N_1, N_2 \ldots, N_k\}$，则通过测量待测链路探测到每个区域发生故障的概率为P_i,设P_i和区域规模成正相关关系，则区域检测的代价C_z可表示为：

$$E(C_z) = \sum_{i=1}^{k} (P_i * C_i) \tag{3-9}$$

C_i是检测可疑区域的代价，事实上有$C_i \geq |N_i|$，这里假设在理想情况下$C_i = |N_i|$。则各区域间检测代价有如下关系：

$$|N| = \sum_{i=1}^{k} N_i = \sum_{i=1}^{k} C_i \tag{3-10}$$

则有：

$$\frac{E(C_z)}{|N|} = \sum_{i=1}^{k} (P_i * \frac{C_i}{|N|}) \tag{3-11}$$

因为$\sum_{i=1}^{k} C_i / |N| = 1$，对任意$P_i > 0$，根据加权均值不等式，可以得到当$P_i$相等时上式取得最小值。由于很难知道区域内待测链路的故障发生概率，但显然不均衡的分区会导致不均衡的待测链路分布，均衡的分割应该是好的选择。

均衡分区的约束要求：$\forall i \in \{1, \ldots, k\}: |Z_i| \leq L_{max} = (1 + \tau)\lceil |N|/k \rceil$。[3]即找到将$G$分成$k$个部分的最小代价分割问题，使每个部分至少包含$(1 + \tau)(n/k)$个节点。

(3)在满足前两个条件的情况下，域间链路尽可能少。因为域间链路并不完全属于某个区域，在基于分区的测量中域间链路是导致漏检的一个重要因素。为了减少漏检，分区产生的域间链路要尽可能少。

条件(2)和条件(3)是衡量图分割的两个经典准则，可以表示为式 3-12：

$$obj(Z) = \begin{cases} minimize(\tau) \\ minimize(cross - edge) \end{cases} \tag{3-12}$$

图分割问题是一个 NP 难的问题[3]，在以往的工作中，研究人员利用启发式和近似算法导出了各种实际的解决方案。均匀或均衡图分割的近似问题被证明是 NP 完全的。

对于复杂网络，采用多层次的方法来对网络拓扑进行均匀的分割具有很好的效果。多层次划分主要包括三个阶段:粗化（Coarsening）、初始划分（Initial Partitioning）和逐级细化（Refinement），具体过程如图 3-4 所示。

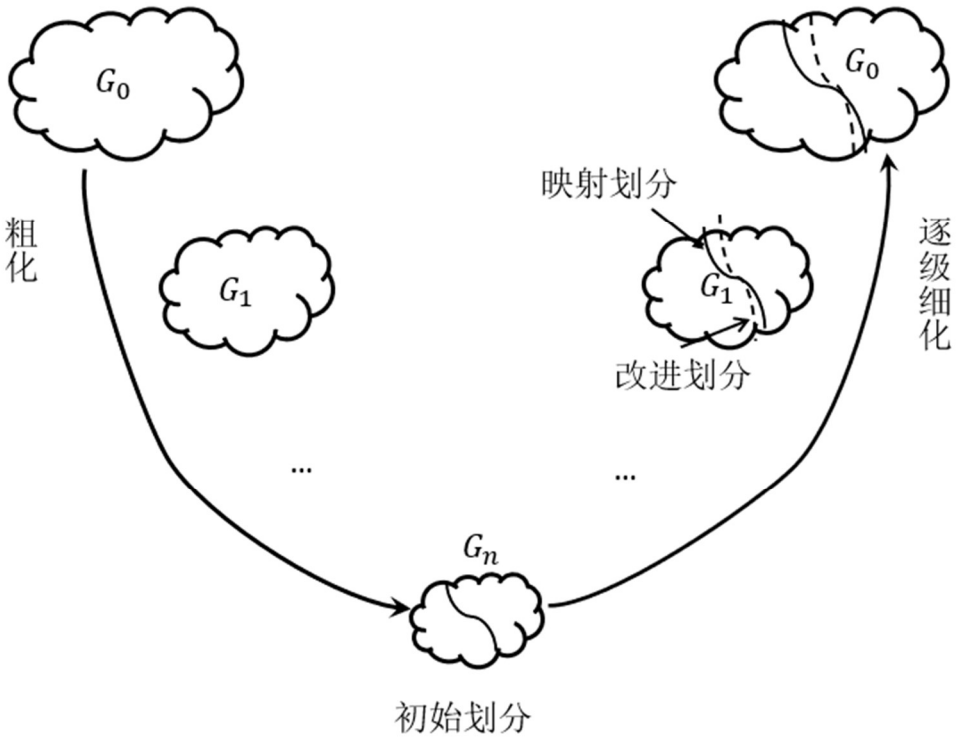


图 3-4 多层次划分的三个阶段

1、粗化阶段：粗化阶段的主要目标是通过成功地创建一系列较小的图来逐步获得输出图。每一个连续图都是由前一个图通过折叠相邻节点对来构造的。这个过程一直持续，直到图的大小减少到只有几个节点。收缩可能产生平行的边，这些边将被单个边取代，单个边的权值累加了平行边的权值。这意味着粗层次上的均衡分区代表了具有相同切割值的细层次上的均衡分区。当图足够小，可以使用任何局部或全局方法进行初始分区时，粗化将会停止。这样的粗化方式保证了对粗化图的分割与对原图的分割产生的割边数相等，均衡的分区也保证了对粗化图的均衡分割可以让原图同样均衡。

2、初始划分阶段：初始划分阶段使用相对简单的方法(如 Kernighan-Lin 算法)来计算最粗图的划分。粗化得到的图通常比较小，所以这一步会很快。在本研究中，使用分级嵌套算法来进行初始划分，主要流程如下：

1）随机选择图的顶点，记为 0；

2）利用广度优先的方法，逐步遍历全图，按级别进行标记。例如 0 节点的邻居被标记为 1，i节点的邻居被标记为$i+1$；

3）当标记节点为总节点数的 1/2，停止。

递归的分级嵌套算法可以有效的对粗化得到的简单图进行均衡的划分。

3、逐级细化阶段：逐级细化阶段包含两步。第一步，将粗化图上得到的解映射到细层次图上。合并在一起的节点对被分配到与其对应的合并节点相同的分区。在每一个投影步骤之后，如果这样的操作提高了分区解的质量，则使用各种启发式方法在分区之间迭代移动节点来细化分区。非粗化阶段结束时，分区解决方案将一路投影到原始图。因为投影分区已经被很好地分割，细化过程将在几次迭代内收敛到一个更好的分区。

多层次的分割方法具有以下优点：1)在粗化过程中的大部分工作已经完成，所以总执行时间不会增加；2) 在初始划分中，单个节点移动对应于最终解决方案中的实质性更改，因此可以很容易地在不增加总体执行时间的情况下找到改进；3) 精细级的局部改进花费时间少，因为初始划分已经获得了较好的分区。

由于图分割算法较为成熟且非本文研究重点，在后续工作中，图分割的基本算法是基于多层划分方法的 METIS 工具生成的，METIS 是现有精度最高的图分割方法之一。METIS 对多级图划分方法的三个阶段都进行了改进，提高了分割的质量。改进的重点是连续减小图的大小，以及在非粗化阶段优化分区。在粗化阶段，使用分级嵌套算法，能够更容易地在最粗的图中找到好的分割。在细化过程中，METIS 的优化重点主要集中在分区边界的部分。

3.3 基于分区的链路传输重要性评价方法

3.3.1 问题描述

根据 3.2.1 中的证明，相对于全覆盖的方案，使用分区的网络测量方案具有更低的代价。为了实现好的分区测量效果，需要选择合理的待测路径集。待测路径集需要能够反映其所在区域状态，换言之，待测链路需要有代表性。但相比于节点，链路具有更多的数量，这意味着链路的选择具有更多的组合，并且不同的链路的故障发生情况是近似独立的，要找到能够满足需求的代表性的链路是一件比较困难的事。从测量角度出发考虑，如果某些链路上所发生的故障占整个区域故障的大部分，这样的链路应该可以作为区域的代表链路。

在通信网络中，链路除了表示连接关系，还具有一个很重要的功能——流的传输功能。网络中的故障，例如最常见的拥塞，通常都是由流的突发性引起的。当一条或多条流同时通过一条链路时，即使流的平均带宽没有超过链路带宽的限制，但由于流的突发性存在，总的带宽可能瞬时超过链路带宽造成拥塞。显然，具有高链路利用率的链路具有更大的发生拥塞的可能性，如果能够取得所有链路的链路利用率快照，就可以将链路利用率作为权值选出待测链路集。然而，链路利用率是一

个动态变量，链路利用率的获得和分析需要额外的代价。为了解决这一问题，本节提出用链路的传输重要性来代替链路利用率作为选取待测链路的方案，传输重要性是一个静态量，当网络拓扑不变的情况下几乎不会发生改变。链路的重要性是根据所有链路在最短路径上的通信频率来衡量的，其中最重要的链路的使用频率是最高的。传输重要性反映的是流通过链路的可能性，传输重要性越高，可能通过该条链路传输的流越多，该条链路发生拥塞的可能性越大。

需要注意的是，网络中的某些路由器可以由端部系统（如主机）直接连接，端部系统可以发送和接收探测包。从一个端系统h_j到另一个端系统h_k的探测报文要经过从h_j到h_k的路由路径，这种端到端路径称为探测路径。在每个探测路径上有两种类型的链路，第一类是终端系统和路由器之间的链路。在互联网中，端系统只能直接连接到边缘路由器，而且它们通常离这些路由器很近，例如在同一栋大楼或校园内。因此，端部系统和边缘路由器之间的链路相当短。第二类是两个路由器之间的连接。由于第一类链路的性能通常比较稳定，在本研究中不考虑评价端系统到路由器之间的链路重要度，也不考虑这类链路发生故障的情况。

3.3.2 基于分区的传输重要性评价方法

实际的网络通信中，数据包通常优先选择一条最短路径进行数据传输，如果该路径出现故障再考虑其他最优路径。在多路径情况下，现有的多路径路由协议如ECMP（Equal-Cost Multi-Path Routing，等价多路径路由），可以选择具有相同路由优先级的多个最佳路径，来实现负载均衡。如上所述，可以看出最短路径中的链路在数据传输中起到重要作用，链路在网络中节点相互通信的最短路径中使用的频度越高，说明这条链路对网络的传输性能影响也越大，重要性也越高。姜[25]提出了一种基于网络传输特性的通信网链路重要性评价方法，根据链路在网络所有节点相互通信的最短路径中的使用频度来评价链路重要性。该方法不需要对链路进行删除和收缩，可以反映链路对网络传输性能直接影响的大小，并且能够评价所有链路，解决了目前算法无法评估收缩和删除链路后的某些链路的问题。

为了适应本文所使用分区的网络测量方案，可以对上述链路重要性评价进行改进。设有网络拓扑$G=(N,\ E)$，采用均衡的图分割方法将网络划分为k个区域（$k\geq 2$）.当网络拓扑进行分区后，可以将链路分为两类，一类是域内链路，一类是域间链路。由于分割的得到的区域内是连通的，可以认为区域内节点之间的传输仅通过区域内部链路，而跨区域的传输还必须通过域间链路。假设端到端之间的通信都是随机的，当拓扑被均衡分割为多个区域，对于单个区域来说，区域内节点的数量总是小于区域外节点的数量，令区域内部的流（源端和目的端都在同一区域内）

的条数为F_{inside}和跨区域流（源端和目的端分别在不同区域）的条数为$F_{outside}$，其期望之比为：

$$\frac{\mathrm{E}\left(\mathrm{F_{outside}}\right)}{\mathrm{E}\left(\mathrm{F_{inside}}\right)}=\frac{1-\frac{|N|}{k}*\frac{1}{|N|}}{\frac{|N|}{k}*\frac{1}{|N|}}=k-1 \tag{3-13}$$

设跨区域路径平均经过链路数为l_{out}，区域内路径平均经过链路数为l_{in}。显然，跨区域路径相对于区域内路径会包含更多的链路，即$l_{out}:\ l_{in}=\alpha>1$。则受跨区域流影响的链路条数和区域内流影响链路的条数之比为$\alpha(k-1)>1$。跨区域流量相对于区域内流量整个网络具有更大的影响，并且随着分区数的增加，影响还会进一步扩大。

由于研究目的不同，本文只需要得到测量所需的待测链路集，不用精确的对比任意两条链路的重要性[25]，而是着眼于选出具有代表性的链路。所以在本文中将不再把区域内部的最短路径纳入统计，而是将跨区域最短路径的使用频度作为传输重要性的重要度。因为根据上述证明，区域间流量对整个网络有较大的影响，使用跨区域路径能够反映出链路的重要度。针对具体的网络，本评价方法只考虑具有流收发能力的节点作为源和目的节点的路径，因为其他节点之间的路径不会影响到经过它的链路的传输能力。

基于分区的传输重要性评价方法如下：

算法

输入：网络拓扑$G=(V,E)$，G的分区$\{Z_1,Z_2\ldots,Z_k\}$，具有流收发能力的节点H

输出：链路$\{e_1,\ldots,e_{|E|}\}$的对应的重要性$\{t_1,\ldots,t_{|E|}\}$

1. 计算不同区域间以H中节点为源和目的节点的最短路径集P
2. 根据P计算链路e_i的使用频度f_i：当某两个节点之间只有一条最短路径时，该路径中所有链路使用频度加 1；当存在n（$n\geq 2$）条路径时，各路径中所有链路使用频度加$1/n$
3. for e_i in E:
4. $t_i=\frac{f_i}{|P|}$ //归一化
5. end for
6. 输出$\{t_1,\ldots,t_{|E|}\}$

该算法和[25]的主要时间复杂度来自最短路径集P的获取，根据 [26]中提出的全节点对的最短路径算法，[25]的算法的时间复杂度为$O(log_2 n*n^2)$。在本算法

中，由于最短路径集只考虑端系统跨区域之间的最短路径，能够减小计算时间

3.3.3 算法示例与对比

对图 3-2 规则拓扑和不规则拓扑示例（b），分别采用[25]中的基于网络传输特性的链路重要性评价方法和前文所述的重要性评价方法对链路进行重要性排序。首先将图 3-2(b)的拓扑按 3.2 中的方法进行分割，得到图 3-5。

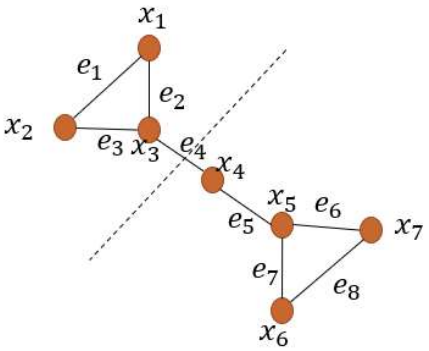


图 3-5 不规则拓扑的分区

根据[25]，所有节点都具有收发能力，本文方法的最短路径集见表 3-1。

表 3-1 基于分区的重要度评价方法的统计路径

路径	源→目的	链路
Path1	$x_1 \rightarrow x_4$	e_2e_4
Path2	$x_1 \rightarrow x_5$	$e_2e_4e_5$
Path3	$x_1 \rightarrow x_6$	$e_2e_4e_5e_7$
Path4	$x_1 \rightarrow x_7$	$e_2e_4e_5e_6$
Path5	$x_2 \rightarrow x_4$	e_3e_4
Path6	$x_2 \rightarrow x_5$	$e_3e_4e_5$
Path7	$x_2 \rightarrow x_6$	$e_3e_4e_5e_7$
Path8	$x_2 \rightarrow x_7$	$e_3e_4e_5e_6$
Path9	$x_3 \rightarrow x_4$	e_4
Path10	$x_4 \rightarrow x_5$	e_4e_5
Path11	$x_5 \rightarrow x_6$	$e_4e_5e_7$
Path12	$x_6 \rightarrow x_7$	$e_4e_5e_6$

基于分区的方法需要统计 12 条路径，而[25]统计G中所有节点相互通信的最短路径，共计C_7^2=21 条路径。

接下来统计每条链路在不同方法的路径中的使用频度f_i，归一化得到重要性t_i。

以e_2为例，e_2在 Path1、Path2、Path3、Path4 中共被使用了 4 次，$f_2 = 4$，归一化后得到$t_2 = 4/12 = 0.333$。注意，两点间若存在n条最短路径（n>1），则统计频度应加$1/n$，而不是 1。两种方法得到的重要度如下表 3-2。

表 3-2 重要度评价方法的对比

链路	基于传输特性的重要性评价[25]	基于分区的传输重要性评价
e_1	0.048	0.0
e_2	0.238	0.33
e_3	0.238	0.33
e_4	0.571	1.0
e_5	0.571	0.75
e_6	0.238	0.25
e_7	0.238	0.25
e_8	0.048	0

从表中可以看出，基于传输特性的重要性评价[25]中得到的链路重要性排序为：$e_4, e_5 > e_2, e_3, e_6, e_7 > e_1, e_8$；而本文采用的基于分区的传输重要度评价方法得到的重要度排序为：$e_4 > e_5 > e_2, e_3 > e_6, e_7 > e_1, e_8$。可以看出重要性的相对大小并没有发生很大的变化，这意味着虽然基于分区的链路重要度评价方法计算和统计了更少的路径，但仍能够有效反映出链路的传输重要性。

由于图 3-5 中拓扑实际上是一个对称的拓扑，在[25]的重要度评价中，拓扑中完全对称的链路，如e_4和e_5、e_2, e_3和e_6, e_7，具有相同的重要度。但在本文提出的基于分区的传输重要度评价方法中，情况发生了变化，$e_4 > e_5$、$e_2, e_3 > e_6, e_7$，这也是两种评价方法得到重要性排序的主要差别，主要原因分析如下：1）在划分区域后，e_4成为了两个区域之间的域间链路，由于基于分区的传输重要性评价方法仅统计跨区域链路，相比于e_5，e_4更多的被跨区域最短路径使用。这种差别对本文的研究是有利的，因为区域内部链路即使没有被待测链路集覆盖，也有机会在对故障区域的二轮中被测量到；但域间链路如果没有被待测链路集覆盖，则只有在域间链路连接的两个区域同时发生故障时才可能被测量到，所以未被待测路径集覆盖到的域间链路被测量到的概率是相对较低的。基于分区的传输特性评价方法产生的差

别提高了域间链路的重要性，使域间链路更可能被第一轮探测覆盖到。2）e_2, e_3和e_6, e_7的差别主要是因为划分区域后，拓扑的对称性被破坏，虽然两个区域具有同样的跨区域最短路径，但由于分割并不完全均衡，区域(x_4, x_5, x_6, x_7)存在更多节点和链路，被最短路径使用的频率相对较少，从而造成差别。在这个例子中，由于重要性的相对大小没有发生很大变化，这样的差别影响并不大，但这也要求了区域的分割应该尽可能的做到均衡。

3.4 基于贪心的探针选择算法

3.4.1 问题描述

在前文中，利用基于分区的链路传输重要性评价可到得到可靠的按分区传输重要度排序的链路。在本节中，将使用重要度按比例选出待测链路集，并由此生成测量探针。

在以往的工作中[59][61][69]，使用主动测量的方案选择探针所提出的架构遵循相同的基本步骤，如图 3-6。

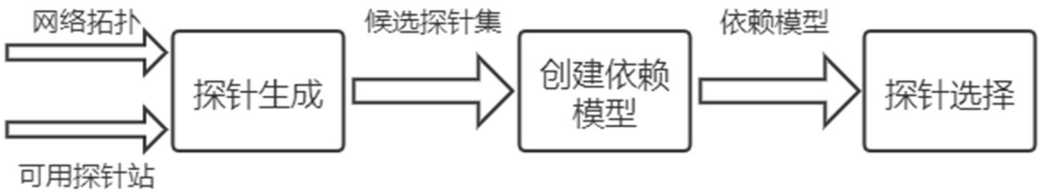


图 3-6 主动测量探针选择步骤

●第一步是探针生成，利用网络拓扑和探测站在网络中的位置生成一组探针，称为候选探针集。候选探测集是一组可用来监视网络的潜在探测集。

●第二步是创建网络的依赖模型。依赖模型存储了候选探针与网络中的节点/链路之间的关系。

●第三步是探针选择，从候选探针集中选择一组探针，称为目标探针集。目标探针集可以检测出与候选探针集相同的故障集合，但理想情况下，目标探针集的大小要小得多。网络的依赖模型有助于选择这样的探测集。

目标探针集通过探针选择算法进行选择，而目标探针集仅限于候选探针集中可用的探针。因此，目标探针集可能不是监控网络的最佳探针集。另一方面，如果使用包含网络中所有可能探测的穷尽候选探测集来选择目标探测集，则会创建较大的网络依赖模型，需要大量的内存来存储和处理。候选探针的大小设置的增加和

依赖模型所需的时间也会增加探测器选择算法选择目标探针集。

上述步骤通常用于全覆盖故障探测的探针选择方案。对于本文来说，由于得到了待测链路集，并且在基于分区的链路传输重要度的运算中得到了跨区域的最短路径集，可以方便的将跨区域最短路径集作为候选探针集，不需要进行第一步的探针生成。

3.4.2 基于待测链路集的探针选择算法

由 3.3 节得到的按重要度排序的链路$e'_1,\dots,e'_{|E|}$，按比例r筛选出最重要的前$\lfloor|E|*r\rfloor$条链路组成待测链路集$\{e_{m1},\dots,e_{m\lfloor|E|*r\rfloor}\}$。具体算法如下：

算法

输入：跨区域最短路径集P，待测链路集E_m=$\{e_{m1},\dots,e_{m\lfloor|E|*r\rfloor}\}$

输出：目标探针集$Probe$

```
while E_m不为空
    Maxp= Null
    Max p_{e_m}= 0   //最大路径信息量，即路径中含有待测链路的数量
    for p_i in P do
        if p_i ∩ E_m 为空 then
            将p_i从 P 移除
        else
            p_{ie_m}=|p_i ∩ E_m|
            if p_{ie_m} > Max p_{e_m} then
                Maxp = p_i
                Max p_{e_m}= p_{ie_m}
            end if
        end if
    end for
    将 Maxp加入Probe
    将 Maxp包含的待测链路从E_m移除
    将 Maxp_i从 P 移除
end while
输出Probe
```

该算法输入跨区域最短路径集P和待测链路集E_m，P作为候选探针集，目的是生成覆盖所有待测链路集E_m的探针集。该算法一直运行到覆盖待测链路集的所有

链路为止。在每一次迭代中，路径选择依赖于链路的信息量，即路径中含有待测链路的数量，选择信息量最大的路径作为探测路径，以减小测量代价。

3.5 仿真分析

为了评估本章所提出的基于分区的重要度评价方法所选择的待测链路集的有效性，本节将对该方法进行简要评估。本仿真暂不考虑具体的故障定位方法，侧重于证明本章所阐述方法的可行性。

3.5.1 对比方案和性能指标

为了检验 3.3 节提出的基于分区的链路传输重要性评价方法的合理性，将随机选择的方案和[25]中的基于传输的链路重要度评价方案作为对比方案。分别用这三种方案筛选待测链路集进行第一轮测量，观察对结果的影响。

除此之外，还设置了不同的待测链路选择比率r、不同的分区数 k 等情况进行仿真，观察结果的变化并进行分析。

本节的实验目的是为了证明所选待测链路集的合理性，将不考虑故障定位和区域测量的具体方法，仿真实验建立在两个假设上：1）假设所有待测链路上发生的故障都能够被检测到；2）检测到待测路径上故障链路后，令其所在的区域为可疑区域，故障区域中发生的故障都能够被检测到。

本仿真的性能指标包括：

1) 待测链路集故障检出率：该指标体现的是待测链路集在实际传输中的重要程度是否和基于分区的链路重要性评价选出的一致。

$$\text{待测链路集故障检出率} = \frac{\text{待测链路集上检出的故障}}{\text{总故障链路数}} \times 100\%$$

2) 总故障检出率：总故障检出数指的是待测链路上发生的故障数加上可疑区域内链路发生的故障数。该指标体现的是待测链路在其所在区域中是否具有代表性。

$$\text{总故障检出率} = \frac{\text{总故障检出数}}{\text{总故障链路数}} \times 100\%$$

3) 链路覆盖率：指的是由待测链路集通过 3.4 中方法生成的测量探针的链路覆盖率。

4) 重要度占比：重要度占比=测量探针覆盖的链路的重要度之和/所有链路重要度之和。

3.5.2 仿真分析

本节采用 pycharm 为仿真工具，使用 python 语言对基于分区的传输重要性的探针选择方案进行仿真。利用 NetworkX 工具来实现随机网络拓扑的生成，利用 METIS 工具来实现均衡的分区。将网络中度小于等于 2 的节点连接主机，主机作为端系统，具有收发能力。在每个周期中，网络中的流量由主机随机生成，传输路径均为最短路径，若源端和目的端之间存在多条路径，应轮询选择多条路径进行传输。当链路中传输的流量大于给定带宽时将该条链路视为故障。每组仿真包含十次实验，每次包含 1000 个周期，取十次仿真的平均作为仿真结果。

对于一个 50 节点的随机网络，将其利用图分割算法划分为 10 个区域，将待测路径比例分别设为 0.1、0.2、0.3、0.4、0.5，仿真结果见后文。

- 仿真一：重要度占比、链路覆盖率和故障检出率的关系

根据图 3-7 可以看出，随着待测链路比例的增加，检出率和待测链路重要度

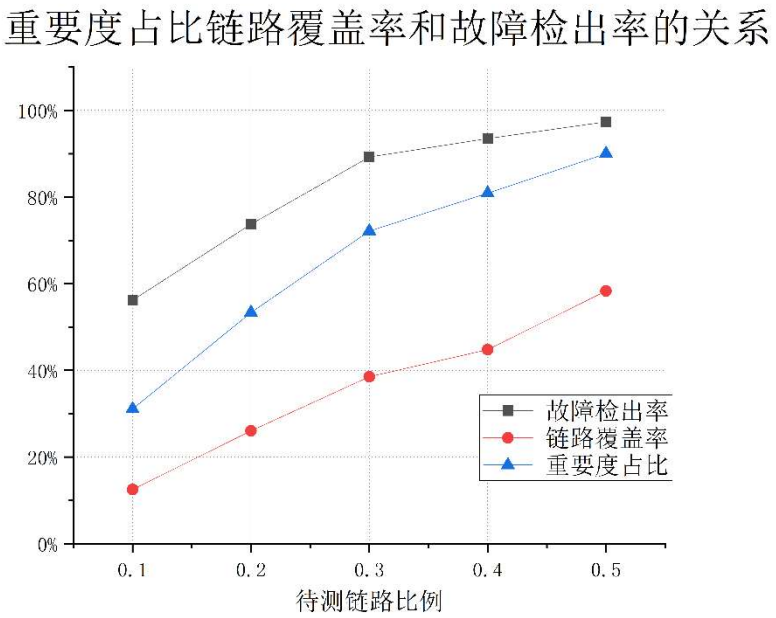


图 3-7 仿真一：待测链路集的重要度占比和链路覆盖率总故障检出率的关系

占比的大致呈相同的趋势，可以说明本文提出的基于分区的传输重要性评价方法的合理性，并且能够用很小的链路覆盖代价得到高的故障检出率。

- 仿真二：区域故障检出率和待测链路检出率的关系

根据图 3-8 可以看出本文提出的基于重要度的待测链路集选择方案相比于随机选择的链路具有很好的故障检出效果，可以说明本文的方法是有一定价值的，能够通过较少的链路检出大部分的故障，而随机选择的方案没有考虑链路的特性，其故障检出率基本和待测链路选择比例成正比。

单独分析本文方案的测量结果，可以看出具有高重要度的链路发生故障的概

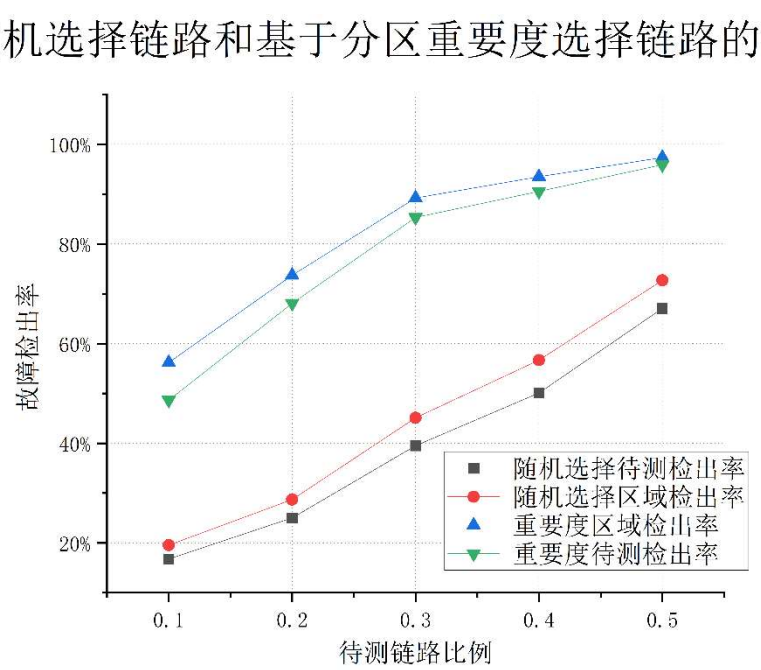


图 3-8 仿真二：随机选择链路和基于重要度选择链路的对比

率占总故障数的很大一部分，其中最高重要度的 10%链路占据了大约一半的故障发生率。随着比例的提高，检出率也随之提高，从 10%到 30%，检出率的变化较大，原因是待测链路占比较小时，存在未被覆盖的区域，会造成整个区域的漏检；随着覆盖率的增加，更多的区域被覆盖，整体检出率呈现明显的提升；当比例达到 30%，区域被待测链路全覆盖，之后再增加链路比例对检出率的提升影响较小。其中 20%待测链路比例的区域和待测检出率具有最大的差值，一是相对于 10%比例，20%比例覆盖了更多探测区域，漏检区域减少，检出率明显增加；二是相对于更大的探测比例，20%比例的剩余的 80%链路中发生故障的可能性更大，更可能在区域测量中被检测到，比例大于 30%之后的差值减小也是同样的原因。

- 仿真三： 不同重要度评价方法的对比

根据图 3-9 可以看出对比方法和本文提出的基于分区的链路重要度评价方法所选出的待测链路具有近乎一致的故障检出率。结合 3.3.3 的算法对比，说明本文提出的基于分区的重要度评价方式虽然减少了计算重要度的统计路径条数，但所得到的重要度排序依然和对比方案一样可靠。

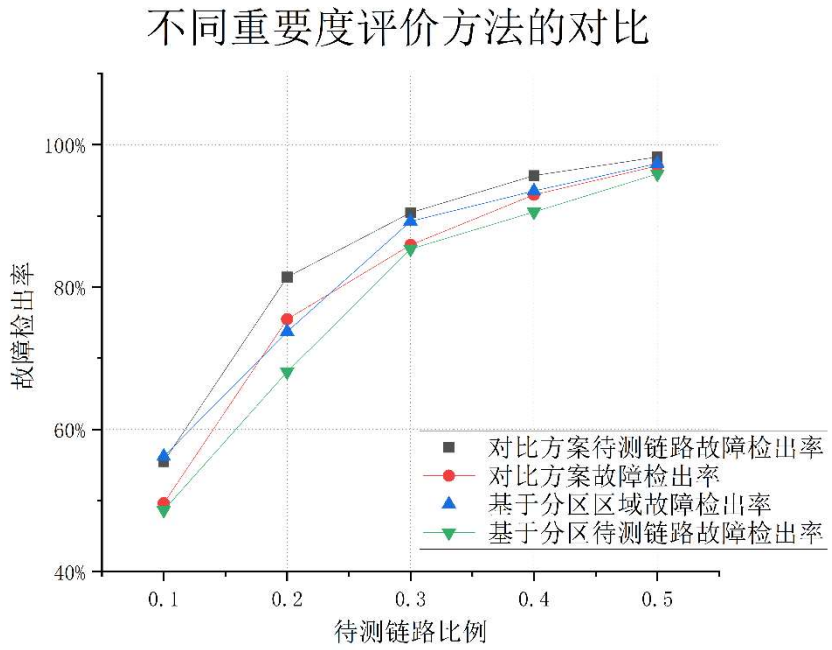


图 3-9 仿真三：本文基于分区的重要度评价方法和对比方法[25]的故障检出率

下面研究分区个数和检出率关系，对于同样的 50 节点的网络，待测链路比例都为 30%，分别划分为 2、4、6、8、10 个区域。

- 仿真四：分区数和检出率的关系

根据图 3-10 可以看出随着区域数的增加，待测检出率变化不大，证明了区域划分对计算链路重要性的影响不大。而随着区域划分，区域检出率的呈现下降的

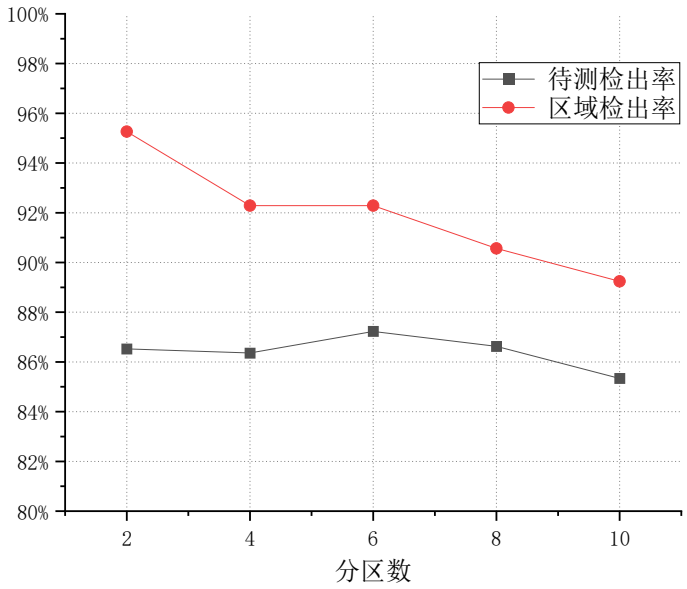


图 3-10 仿真四：分区数和检出率的关系

趋势，原因是，虽然待测链路集变化不大，但分区数越小，区域规模会变大，每个区域内的待测链路变多，检测到故障的概率变大，区域被检测的次数变多，发现区域内故障的可能性变大；同时少的区域分割会带来少的域间链路，更少的域间链路意味着更少的漏检。但大的区域会带来更频繁的二轮检测，同时对区域的检测代价也会变大，在检出率相差不大的情况下，还是应该选择较多的划分。

- 仿真五：分区个数和重要度及链路覆盖率的关系

根据图 3-可以看出区域的个数对重要度占比和链路覆盖率的影响不大，可以认为在这个前提下选择更大的分区个数对待测链路集的选择影响不大，考虑到测量代价，应该选择尽量大的分区个数。

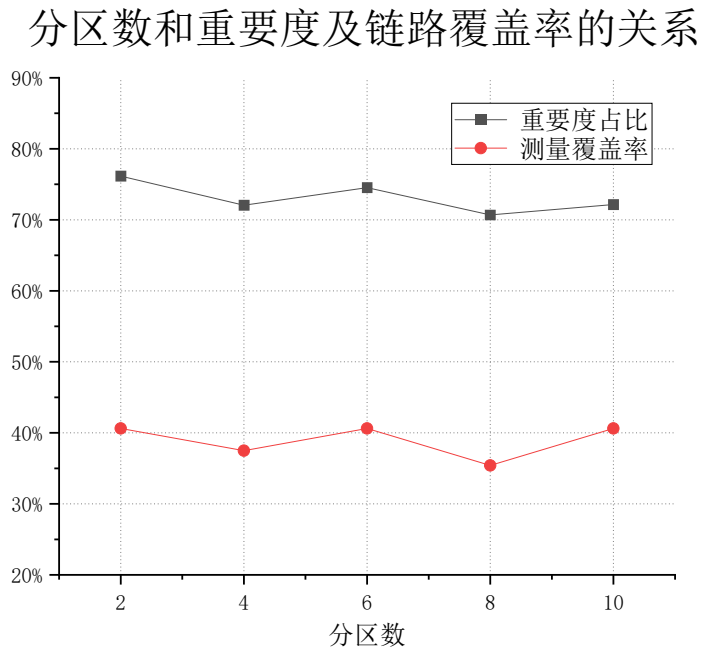


图 3-11 仿真五：重要度占比和链路覆盖率与分区数的关系

3.6 本章小结

本章首先阐述了拓扑图分割的相关研究，证明了相比于链路全覆盖的测量方案，基于分区的链路故障测量方案在测量代价上有优势；然后对具体的拓扑分割方法进行了阐述。然后探讨了基于分区的链路重要度评价方法，并给出了示例和与对比方案的比较；接着对根据重要度筛选出的待测链路集的探针选择算法；最后仿真分析了在不同待测链路选择比例和分区个数下的故障检出率，并且与随机选择和使用对比方案选择的待测链路集的故障检出率进行了对比。

第四章 可疑区域的定位和探测

4.1 问题描述

通常链路级故障检测的目标是检测网络链路上可能发生的任何性能下降或基础设施故障。在[82]中提出了建立的可分性能特性的可分故障，当且仅当路径至少有一个连接故障时，路径显示可分离故障。故障检测过程传递的信息是一组故障路径，即显示故障的探测路径。将仅由故障监控路径所遍历的链路集称为可疑链路集。仅凭检测信息无法判断这些链路是否异常。

现有的网络监控方案几乎都采用两步法进行监测位置和探测路径的选择。通常，第一步是选择能检测/定位所有潜在异常的最小探测位置集用于安放探针站。第二步选择第一步所选探测点之间的最小路径集，以覆盖或区分所有潜在异常。查明检测到的异常的源头而需要监控的路径集有两种定义方式。第一种方法是监视一组路径，该路径可以区分每一对链路级异常对于任何检测到的异常；第二种方法则是监视一组选择的路径，该路径在检测到异常时只能区分一组可疑链路。探测器的位置问题和路径选择问题都是 NP-Hard 问题。很多的启发式算法被提出用来解决此问题，其中大多数采用贪婪算法。

在本文中，利用第三章的探测路径选择算法，得到了待测链路集和测量探针。向待测路径发送探针进行周期性的主动测量（本文将之称为第一轮探测），能够得到探测路径的路径状态。但这些数据只能反映路径级的故障状况，并不能从故障检测探针的测量数据中定位到具体故障链路。并且由于探针包含的链路有限，无法检测到网络中所有的潜在故障。需要利用测量探针返回的结果进一步分析来尽可能的得到多的故障信息。一旦探针返回的结果体现出路径异常，无法获知未被覆盖的链路信息，为了了解链路级的故障情况，本章利用探测得到的链路信息将可疑链路进行分类，通过构造额外的诊断探针，定位得到探针覆盖链路中的异常链路集。将存在异常链路的区域称为可疑区域。第二轮测量主要目标就是对可疑区域中未被第一轮测量覆盖的链路进行进一步测量。

4.2 可疑区域定位

4.2.1 问题建模

为了得到路径和链路的状态关系，可以将每个测量路径建模为它所遍历的链路序列。当一个包正确地到达它的最终目的地时，认为它所经过的链路是正常的。

另一方面，当数据包在传输中出现异常时，如传输时延过长或发生丢包，说明数据包传输经过的路径出现了故障。当路径中至少有一条链路出现故障时，就会出现后一种情况。通过对测试路径测量结果的观察归纳出一个布尔方程系统，其中未知的是网络中链路的布尔状态。这个方法主要的挑战是由这种布尔系统的不确定性带来的——多个故障场景会导致相同的观察路径探测结果。为了得唯一标识n条链路的状态，需要构造n个对应的线性独立测量值。但由于网络中的故障链路常常是稀疏的，可能需要大于n次的测量才能构成完整的测量矩阵。

将网络建模为$G = (V, E)$，其中节点集V表示网络节点，边集E表示链路。使用$P_{s,t}$表示从源节点s到目标节点t所经过的路径。令$\mathcal{P}$是所有探测路径的集合，所对应的路由矩阵为D，如果路径$P_{s,t} = P_i$包含链路l_j，D_{ij}= 1，反之为 0。因此，D 的一行对应于路径，而列对应于链路。

如果一个路径至少包含一个故障链路，那么它就被定义为故障;如果一个路径中不包含故障的链路，那么它就被定义为“健康的”。我们使用y_i来表示路径P_i的状态:如果P_i是健康的，y_i= 0，否则y_i= 1;使用x_k表示链路l_k的状态，如果x_k是健康的，则x_k= 0，否则x_k= 1。可以建立一个关于y_i和x_k的布尔代数线性方程组:

$$y_i = \vee_{k=1}^{n_c} x_k \cdot D_{ik} \tag{4-1}$$

其中"∨"表示二进制加法运算，"·"表示乘法运算。以上公式表明了测量得到的路径状态和对应链路状态的关系，利用探测路径和其经过链路的关系可以进一步推断故障范围。

4.2.2 故障路径中的故障链路定位

本节以一个具体的网络拓扑为例，阐述如何对故障探测得到的故障路径进行进一步的故障源定位。

随机生成一个 20 节点的网络，以该图为例，将网络拓扑进行图分割，节点所在区域Z_i和图中节点颜色的对应关系为：Z_1-红、Z_2-蓝、Z_3-绿、Z_4-黄。通过第三章所述的探针选择方法（r=0.3）计算出待测路径，拓扑和待测路径如图 4-1 所示。

图 4-1 中红色边为探测路径，具体的一轮探测路径如**错误!未找到引用源。**所示。可以看出在r=0.3 的待测比例下，一共有 5 条一轮探测路径，覆盖了 11 条链路以及所有 4 个区域。

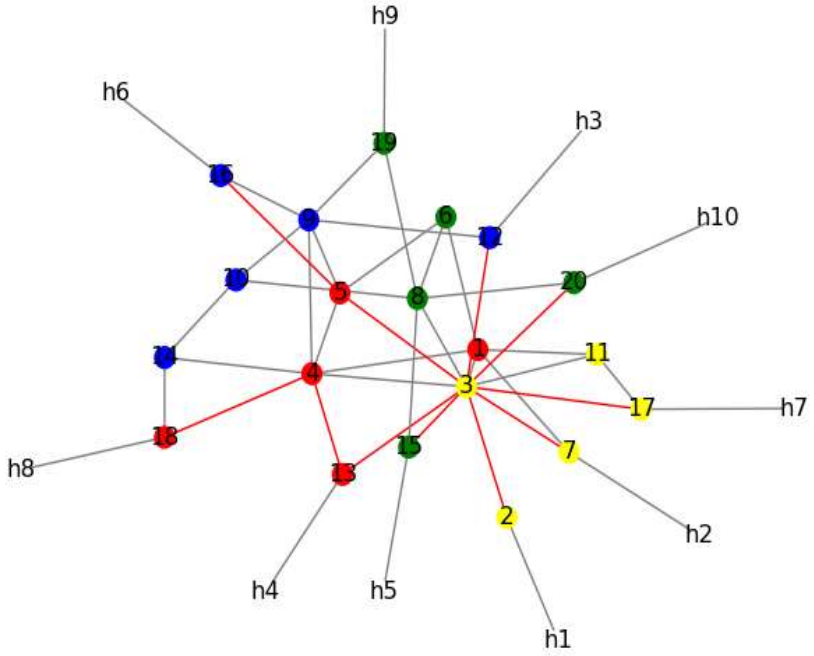


图 4-1 作为示例的 20 节点的网络

表 4-1 一轮探测路径

路径	链路		
p16	(2,3)	(3,5)	(5,16)
p23	(7,3)	(3,12)	
p45	(13,3)	(3,15)	
p48	(13,4)	(4,18)	
p710	(17,3)	(3,20)	

在[83]中给出了可识别的定义：当其中一个环节出现异常时，如果能够确定哪一个环节是异常的，那么这两个环节就是可识别的。根据 4.2.1 中得到路径和链路的关系，尝试对链路状态进行识别。观察表 4-2 所示测量路径矩阵$\boldsymbol{D}$，由于各探测路径之间不存在相同的链路，并不能通过已有探测路径区分各路径上的链路状况，但是由于探测路径经过相同的节点，可以考虑通过添加具有公共链路的测量来实现对故障路径中链路状态的判定。在本文中，称用于区分故障链路状态的路径为诊断路径，用符号$p_{diagnose}$表示。

表 4-2 $\boldsymbol{D}$：探测路径和链路的关系

$\boldsymbol{D}$	(2,3)	(3,5)	(5,16)	(7,3)	(3,12)	(13,3)	(3,15)	(13,4)	(4,18)	(17,3)	(3,20)
p16	1	1	1	0	0	0	0	0	0	0	0
p23	0	0	0	1	1	0	0	0	0	0	0
p45	0	0	0	0	0	1	1	0	0	0	0
p48	0	0	0	0	0	0	0	1	1	0	0
p710	0	0	0	0	0	0	0	0	0	1	1

诊断路径的构造思路通过一个具体例子来说明：假设在一个时间间隔内链路状态不会发生改变，当检测到 p23 发生故障，根据链路和路径的关系，可以推断 p23 上的链路{(7,3),(3,12)}至少存在一条故障链路；p45 没有检测出故障，称为健康路径，那么其上链路{(13,3)、(3,15)}都是健康的。为了识别{(7,3),(3,12)}的状态，可以构造新路径 p25：{(7,3)、(3,15)}，如果 p25 为健康的，那么根据链路路径状态关系可知，(7,3)为健康的，而 p23 中至少存在一条故障链路，所以(3,12)必为故障；如果 p25 为故障的，那么根据链路路径状态关系可知，(7,3)为故障，但不能确定(3,12)的健康状况，可以继续构造路径如{(13,3),(3,12)}来获得其状态信息。因为网络中故障总是稀疏的，可以合理假设总是存在健康路径用以构造诊断路径。

通过对$\boldsymbol{D}$的进一步分析，可以看出有一些链路不能构造出对应诊断路径用以分辨，例如{(3,5),(5,16)}和{(3,4),(4,18)}，这些链路被称为不可识别的待测链路。事实上，只有当诊断路径加入后使对应链路的列在$\boldsymbol{D}$中唯一，才能唯一确定该链路的状态。解决这个问题可以通过添加新的探测路径来使列唯一，例如添加 p78{(3,17),(3,4),(4,18)}，就可以区分{(3,4),(4,18)}的状态。本测量方案不采用添加探测路径的方法，原因是所选择的待测链路是作为代表性链路代表它所在区域的状态，在检出待测链路存在故障时本就需要继续确定其所在区域内链路的状态，所以可以将无法区分的链路看作一个整体，其对应的区域都看作是可疑区域，在二轮探测中再做处理。在这个方法下，可以将无法识别的链路合并，将D简化为D'，如表 4-3 所示。

$\boldsymbol{D}'$中对应列对应的元素称为最小可识别集s_i, s_i是一条或多条链路的集合。 最小可识别集可以通过探测路径的交点来获得。将探测到的故障的最小可识别集称为最小故障集，最小故障集s_{fault}需满足以下条件：

$$\exists e_m \in s_{fault}\ ,\ \ e_m \in L_{fault} \tag{4-2}$$

L_{fault}是所有故障链路的集合。只要存在故障链路，该最小可识别集即为最小故障集，只有当其中所有故障链路都是健康的，最小可识别集才为健康的。

表 4-3 $\boldsymbol{D'}$：最小可识别集和测量路径的关系

$\boldsymbol{D'}$	(2,3)	(3,5) (5,16)	(7,3)	(3,12)	(13,3)	(3,15)	(13,4) (4,18)	(17,3)	(3,20)
p16	1	1	0	0	0	0	0	0	0
p23	0	0	1	1	0	0	0	0	0
p45	0	0	0	0	1	1	0	0	0
p48	0	0	0	0	0	0	1	0	0
p710	0	0	0	0	0	0	0	1	1

根据上面的分析，本文归纳提出了一种选择诊断路径进行最小故障集定位的算法。该算法选择需要通过网络发送的额外的诊断探针，以执行更深入的定位。当第一轮测量探测到故障时，将得到故障路径集作为输入。通过已知的健康路径得到健康链路集，利用健康链路集构造最短的诊断路径$p_{diagnose}$，然后利用$p_{diagnose}$上发送的探针结果判断最小可识别集的状态。

诊断故障链路的具体算法如下：

算法

输入：网络拓扑G，探测路径集$\mathcal{P}$，故障路径集P_{fault}

输出：故障的最小可识别集S_{fault}

1. 健康路径集$P_{health} = \mathcal{P} - P_{fault}$，对应健康链路集$L_{health}$
2. $for\ p$ in $P_{fault}\ do$
3. p对应的最小可识别集的集合S_p
4. $for\ i$=0 to $|S_p|$
5. 构造含有S_p中第i个最小可识别集s_i的所有链路，且其余链路在L_{health}上的路径，作为诊断路径，选出最短的诊断路径$p_{diagnose}$。
6. 沿 $p_{diagnose}$发送探针，分析探针返回的结果
7. $if\ p_{diagnose}$故障：
8. s_i故障，加入S_{fault}
9. $end\ if$
10. $if\ P_{diagnose}$健康：
11. s_i健康，将s_i中的链路加入L_{health}
12. $end\ if$
13. $end\ for$
14. $end\ for$
15. 输出S_{fault}

4.3 可疑区域的探测

得到故障链路集后，根据故障链路集中所包含的链路，本文将这些链路两端节点所在的区域称为可疑区域。在图 4-1 中，由于在待测链路测量阶段拓扑经过分割，每个区域的规模是比较小的，可以考虑对可疑区域内链路进行全覆盖的探测。在本文中称为第二轮探测。

和第三章所述的基于重要度的探针选择方案相比，全覆盖的方案更注重探测路径的选择。探测路径的选择是基于探测的网络链路监测的主要问题。一般来说，有两个重要的考虑因素，即最小化探测代价和实现可识别性。探测代价主要定义为所选择的探测路径的个数。也可以是探测的端系统个数，或者是由网络组件定的开销等。由于探测流量周期性地注入到网络中，消耗了网络带宽，增加了路由器的工作量。探测成本越低，资源消耗越少，对正常数据传输的负面影响越小，探测方案的可扩展性越好。所以应该尽量避免选择冗余的探测路径，来减小测量代价。

在这一节中，首先分析了全覆盖方案的测量代价，然后基于本文之前的工作提出了对可疑区域的基于启发式的贪婪算法。

4.3.1 测量代价分析

如前所述，全覆盖方案的主要思路是，当且仅当路径至少有一个链接异常时，路径显示可分离异常。从这个属性可以得出一个简单的结论:为了检测给定网络中所有潜在的链路级异常，必须监视覆盖网络所有链路的一组路径。如果至少有一条监视路径穿过一个链接，就称该链接为覆盖的。这是链路级异常检测的充分必要条件。

现有的链路级故障检测大多两步进行：第一步是选择能检测/定位所有潜在链路级故障的最小监视位置集。第二步选择所选监视位置之间的监视路径的最小集，以便成对覆盖/可区分网络的所有链接。这两步都会产生对应的测量代价。在全覆盖的测量中测量的代价主要包含以下几类[82]：

- 基础设施代价：表示获取、部署和维护软件和硬件监控设备的代价。让C_{infra}承担在网络节点上安装和维护一个监控设备的成本。设Y_n为一个二进制变量，表示是否选择节点 n 作为探针站位置。基础设施成本可以表示为:

$$C_{infra}\sum_{n\in N}Y_n \tag{4-3}$$

基础设施代价的最小化是为了部署尽可能少的监控设备。

- 链路探测代价：该代价表示主动测量方法向网络中注入探针产生的额外负载。在全覆盖方案中，网络中的每条链路都必须至少有一条监控路径进行监控。然

而，冗余的测量是非常不可取的。这是因为它们只增加了检测延迟和开销，但不能提供任何额外的检测信息。M_p是一个二进制变量，它指示是否选择要监视的路径p。设δ_{pe}为一个二进制输入参数，表示路径p是否覆盖链路e。设C_e为沿链路e注入一个探测流的代价。一个链路 e 被测量的次数等于测量探针穿过e的次数，表示为$\sum_{p\in P}\delta_{pe}M_p$。总的链路测量代价表示如下：

$$\sum_{e\in E,p\in\mathcal{P}} C_e\,\delta_{pe}M_p \tag{4-4}$$

- 探针发送代价：该代价表示了主动测量方法向网络中发送探针产生的发包代价。探测包的生成和发送是有代价的，源端需要负责探针的生成，包括源和目的端的信息，并根据探测包的具体功能添加需要的信息。例如，测量端到端延迟就需要向探针中加入发送时间戳。S_n是探针站在一个探测周期发送的探针数量，探针发送代价可以表示如下：

$$\sum_{n\in N} Y_n S_n \tag{4-5}$$

- 路由转发代价：该代价表示了探测包在网络中进行传输产生的转发代价。最简单的转发代价和探针经过的跳数有关，当一个探针所经过的节点越多，其产生的转发代价会越高。设δ_{pn}为一个二进制输入参数，表示路径p是否经过转发节点n。设C_r为一个探针被转发一次的代价。路由转发代价可以表示如下：

$$\sum_{n\in N,p\in\mathcal{P}} C_r\,\delta_{pn}M_p \tag{4-6}$$

除此之外根据具体条件不同的还会产生其他代价，如 SDN 环境下的交换机流规则代价，和网络管理中心交换测量结果的通信代价等，需要结合具体情境分析并考虑。

在本文考虑的测量条件中，探针站基于已有的端系统设置，因此额外的基础设施代价不被具体考虑，并且探针发送和转发产生的代价对于现在的网络设备来说几乎可以忽略不计。由于本文的整体方案突出考虑的是减小测量注入的探测流量，所以重点考虑如何将链路测量代价最小化。

为了实现全覆盖，需要保证每条链路e都至少被一条探测路径覆盖，这被称为全覆盖约束，表示为：

$$\sum_{p\in\mathcal{P}} \delta_{pe}\,M_p \geq 1,\ \forall e\in E \tag{4-7}$$

4.3.2 可疑区域分析

由于在第三章中已经得到了确定的分区和测量路径，通过 4.2 节的分析也确定了能够通过被探针覆盖链路确定的最小可识别集。当最小可识别集被定位到存在故障，其对应区域被称为可疑区域，记为S。可疑区域 S 的充分必要条件是最少存在一条链路是异常的，可以表示为：

$$e_m \in E_s，\ if\ \exists e_m \in L_{fault} \tag{4-8}$$

E_s是可疑区域包含的所有链路组成的链路集。由于一条链路连接两个节点，并且还有含有多条链路的最小可识别集的存在，可疑区域实际上是拓扑分区$\{Z_1, Z_2 \ldots, Z_k\}$中一个或多个元素所组成的集合。以图 4-1 为例，利用 4.2.2 节中得到的最小可识别集，通过将最小可识别集中所包含的节点和其所在的区域进行对应，能够得到最小可识别集和对应区域的关系，如表 4-4 所示。

可以看出，同一个可疑区域中可能包含多个最小可识别集。考虑多个故障的存在，将所有最小可识别集的对应故障区域进行组合可以得到所有可能的可疑区域集：$\{Z_1, Z_4，\ Z_1Z_4，\ Z_2Z_4，\ Z_3Z_4，\ Z_1Z_2Z_4，\ Z_1Z_3Z_4，\ Z_2Z_3Z_4，\ Z_1Z_2Z_3Z_4\}$。

表 4-4 最小可识别集和可疑区域的关系

最小可识别集	(2,3)	(3,5) (5,16)	(7,3)	(3,12)	(13,3)	(3,15)	(13,4) (4,18)	(17,3)	(3,20)
可疑区域集	Z_4	$Z_1Z_2Z_4$	Z_4	Z_2Z_4	Z_1Z_4	Z_3Z_4	Z_1	Z_4	Z_3Z_4

可疑区域中的链路可以分为两类：第一类是已被探测路径集覆盖的链路；第二类是未被任何探针覆盖的链路，这也是本轮全覆盖探测需要测量的主要目标。第一类中的链路又可以分为状态已确定的链路和状态未确定的链路。链路状态的确定与否和链路的可识别性息息相关，可以通过以下条件加以判别：

1） 如果最小可识别集只包含一条链路，那么无论其故障与否，其中的链路状态都是确定的。

2） 如果最小可识别集包含多条链路，其状态为健康，则其中的链路状态都是确定的。

3） 如果最小可识别集包含多条链路，其状态为故障，则其中的链路状态是不确定的，并且至少存在一条故障链路。

综上，由于状态未确定的链路中一定存在故障链路，所以在第二轮探测后需要将这些链路加入二轮探测的可疑路径集进行进一步处理。接下来将讨论可疑区域探测的探针选择。

4.3.3 二轮探测路径选择

全覆盖方法探测路径选择的基本思想是选择最有用的探测路径，避免选择冗余的探测路径。仍以图 4-1 为例，由于红色链路是一轮测量中被探测路径覆盖的链路，并且其所在的最小可识别集的状态已确定。显然，如果在第二轮测量中继续覆盖红色链路会带来不必要的代价，在本节提出一个启发式的贪心算法作为测量路径的选择。

现有的探测路径选择一般分为两步，第一步选出满足一定要求的候选路径集，第二步通过算法选择最符合要求（通常是信息量最大）的路径加入探测路径集，直到所有链路都被覆盖或没有满足要求的路径为止。需要注意的是，候选路径集的生成需要符合路由规则，如最短路径，如果为待测路径专门规划路由，显然会带来不必要的计算和维护成本，但由于这种限制的存在，可能导致产生链路无法被覆盖的情况，通常将这样的情况称为漏检。对于漏检，常常采用的方法是增设探针站，以引入能够覆盖漏检链路的探测路径。但在本文的研究场景中，探针站只能基于已有的端系统设置，在这样的假设下，部分链路的漏检是不可避免的，所以本节提出的全覆盖方法，实际上是一种尽最大能力的覆盖。

本节所使用的覆盖方法首先将通过最小故障集得到的可疑区域集作为输入，通过可疑区域集中的所有链路导出的子图即为二轮测量的拓扑，候选探测路径集为该拓扑和其中的端系统之间的所有最短路径的集合。注意，为了尽量避免对一轮测量覆盖的链路进行冗余测量，将这些链路的权值增大并进行加权最短路计算。令可疑区域中的所有未被一轮测量覆盖的链路为目标链路集,选择信息量最大的候选路径加入二轮探测路径集，直到所有链路都被覆盖或没有满足要求的路径为止。

以图 4-1 为例，该图总链路数为 36，也就是说全覆盖代价C_{all} ≥36，根据上述算法计算得到各可疑区域集对应的测量代价，如表 4-5 所示。

观察表 4-5 可以发现，只有可疑区域集包含全部区域时，总探测代价大于 36。考虑到故障的稀疏性，所有区域同时为可疑区域的可能性是很低的，也就是说本研究方法的两轮探测总代价总是比直接全覆盖的代价低的。关于测量代价有关的实验将在第 5 章进行。

通过第二轮测量可以得可疑区域中的探测路径的测量结果，将二轮探测得到故障路径和 4.3.2 节所讨论的无法确定状态的最小故障集一同设为可疑链路集，子图中的剩余链路为健康链路，进一步对故障进行定位，定位的方法和对 4.2.2 中对第一轮探测的故障链路定位方法相同，在此不再赘述。

表 4-5　图 4-1 的可疑区域测量代价

可疑区域集	总探测代价	漏检链路
无	11	\
Z_1	11	2
Z_4	14	0
Z_1Z_4	22	1
Z_2Z_4	18	2
Z_3Z_4	21	0
$Z_1Z_2Z_4$	28	2
$Z_1Z_3Z_4$	32	2
$Z_2Z_3Z_4$	27	1
$Z_1Z_2Z_3Z_4$	37	3

4.4 本章小结

本章首先探讨了链路故障和路径故障之间的关系；随后以一个具体的拓扑为例，阐述了对第一轮的探测路径进行故障定位的方法，提出了最小可识别集的概念；随后讨论了第二轮测量的测量代价，分析了最小可识别集的链路状态和其对应的可疑区域，提出了可疑区域的待测链路选择算法。并对示例拓扑进行了测量代价分析。

第五章 多步分区主动测量方法的仿真实验和结果分析

在第三章和第四章具体阐述了本文提出的基于分区的探测方法并以具体例子进行了详细说明。该方法包括两轮探测，第一轮对基于链路的重要度排序选出的待测链路集进行了测量，第二轮通过对第一轮探测中得到的故障路径分析和定位，得到了最小故障集和其对应的可疑区域，并对可疑区域中未被覆盖的链路进行了第二轮测量。可以合理猜测在这两轮的测量代价存在类似负相关的关系，可能会存在某一个均衡点使总的探测代价最小。本章为了体现本文研究方法的在不同拓扑模型和流量模式下适用性，将通过广泛的实验来寻找各类条件下的最优参数和最低代价。本章还考虑了如何利用额外的链路信息对测量进行进一步的优化。

5.1 拓扑模型

为了更好的理解网络拓扑结构和网络中的流量行为等关系，需要对实际的网络拓扑结构特征有深入的了解，从而建立适合的网络拓扑模型。自二十世纪 60 年代开始，对复杂网络的研究主要集中在随机网络上。最典型的随机网络是 ER 随机图，1998 年 Watt 和 Strogatz 提出了小世界网络，1999 年 Barabas 和 Albert 提出了无标度网络，之后涌现出了越来越多的针对实际网络拓扑特征的各类模型。限于篇幅，本节将简要介绍本章实验中将用到的以上三类典型的随机网络模型。

本文中所有网络拓扑都使用 NetworkX 生成。NetworkX 是一个 Python 包，用于创建，探索和分析研究复杂网络的结构，动态和功能。NetworkX 图中的节点可以是任何 Python 对象，并且边缘可以包含任意数据；这种灵活性使 NetworkX 成为代表许多不同科学领域中发现的网络的理想选择。除了基本的数据结构外，许多图形算法还用于计算网络属性和结构度量：最短路径，中心性，聚类和程度分布等等。NetworkX 可以读写各种图形格式，并提供许多经典图形和流行图形模型的生成器。

本文所提出的探测方案是基于 IP 上层完成的，对于底层的路由规则及数据包转发过程并不敏感。因此在仿真采用网络模型选取时更多考虑的是模型的固件成熟度、灵活性以及代表性，主要目的在于验证所提出方法是否各类拓扑都具有良好的适应度和通用性。而在本文提出的分步探测定位方法的具体应用过程中，对网络拓扑的参数设定和属性并无限定或特殊要求，后续的仿真实验也证明了这一点。

5.1.1ER 随机网络

在图论中，ER 随机图（Erdoes-Renyi random graph）是一种网络，以概率p连

接N个节点中的每一对节点。ER 模型随机网络有一个重要特性：虽然节点之间的连接是随机形成的，但最后产生的网络的度分布是非常均衡的。在网络中，每个节点都与另外某些节点直接相连，节点直接连接的邻居数目叫做节点的度。节点的度的概率分布称为节点的度分布。度分布表明节点的度的分布情况。

在 ER 模型中，大部分的节点的度都集中在一个特定值附近，呈钟形 Poisson 分布，远大于或远小于这个值的概率极小。

利用 NetworkX 生成 ER 随机网络的方法如下图，p表示两点之间的连接概率。

```
1  import networkx as nx#导入NetworkX模型包
2
3  #生成一个有50个节点，连接概率为0.1的ER随机网络
4  er=nx.erdos_renyi_graph(50,0.1)
```

图 5-1 生成 ER 随机网络

5.1.2 小世界网络

Watts 和 Strogatz 提出了 WS 小世界模型，这种模型是一类特殊的复杂网络结构，是完全规则网络到完全随机网络的过渡模型，在这种网络中大部分的节点彼此并不相连，但绝大部分节点之间经过很少的跳数就可到达。

小世界网络的典型特征式具有较小的平均长度且聚类系数较高。由于较高的聚类系数，小世界网络中不可避免地会有许多团（一小群相互连接的节点）以及只比团差几个连接的节点群。另外，任意两个节点至少存在一条短路径，这是有小的最短路径长度平均值带来的结果。

利用 NetworkX 生成小世界网络的方法如图 5-2，概率p反映网络的随机情况，当p=0.1，接近规则网络；p=0.9，接近随机网络：

```
1  import networkx as nx#导入NetworkX模型包
2
3  #生成一个有50个节点，每个节点有5个邻居，p=0.5的小世界网络
4  ws=nx.watts_strogatz_graph(50,4,0.5)
```

图 5-2 生成 WS 小世界网络

5.1.3 无标度网络

在网络理论中，无标度网络（Scale-free network，也称 BA 网络）是带有一类特性的复杂网络，现实中的许多网络都带有无标度的特性，例如万维网、因特网、

金融系统网络、社会人际网络等等。

无标度网络的特性，正是在于其度分布没有一个特定的平均值指标，而是服从 Pareto 分布。该模型基于实际网络中普遍存在的“增长（growth）”和“偏好连接（preferential attachment）”两个重要特性来构造。其典型特征是在网络中的大部分节点链接数很少，而有极少的节点与非常多的节点连接。

利用 NetworkX 生成小世界网络的方法如图 5-3 所示。

```
import networkx as nx#导入NetworkX模型包

#生成一个有50个节点，每次加入两条边的无标度网络
ba=nx.barabasi_albert_graph(50,2)
```

图 5-3 生成 BA 无标度网络

5.2 流量模式

网络中业务的空间分布情况和链路的故障发生息息相关。许多流量模式（Traffic Pattern）被用来反映的空间分布，如最近邻（NN），均衡随机（UR），位补码（BC），热点（HP），龙卷风（TOR）等，这些流量模式常用于评估网络拓扑和路由算法的性能。

流量模式可以按其特性的不同分为良性流量模式和对抗性流量模式。良性流量模式，如 NN 和 UR，具有天然负载均衡性，意味着其中的故障分布会相对均匀；对抗性流量模式，如 HP 和 TOR，会导致不均衡的业务负载，这意味着某些链路更容易成为瓶颈链路。本章采用随机生成的复杂网络拓扑，不适用某些用于规则拓扑的流量模型，使用的流量模式如表 5-1 所示。

表 5-1 流量模式介绍

流量模式	
UR	均衡随机(Uniform Random):每个节点发送到一个随机选择的节点。
HP	热点（Hotspot）：每个节点以相同的概率将消息发送到其他节点，但特定节点（称为热点）以更高的概率接收消息。例如，热点相比于其他节点多 10%的概率。
TOR	龙卷风（Tornado）：结点 i 将分组发送给结点$(i + \lceil k/2 \rceil) mod\ k$，其中$k$为端系统总数，根据网络拓扑具体计算。

5.3 仿真分析

5.3.1 实验拓扑

使用 NetworkX 工具生成具有 50 个节点的 ER 随机网络、WS 小世界网络、和 BA 无标度网络，通过修改不同的参数使图中的链路数从小到大增加，生成的拓扑和具体参数如图 5-4 所示。每类模型通过改变参数生成了 3 个链路数量依次增加的拓扑，共 9 个实验拓扑。

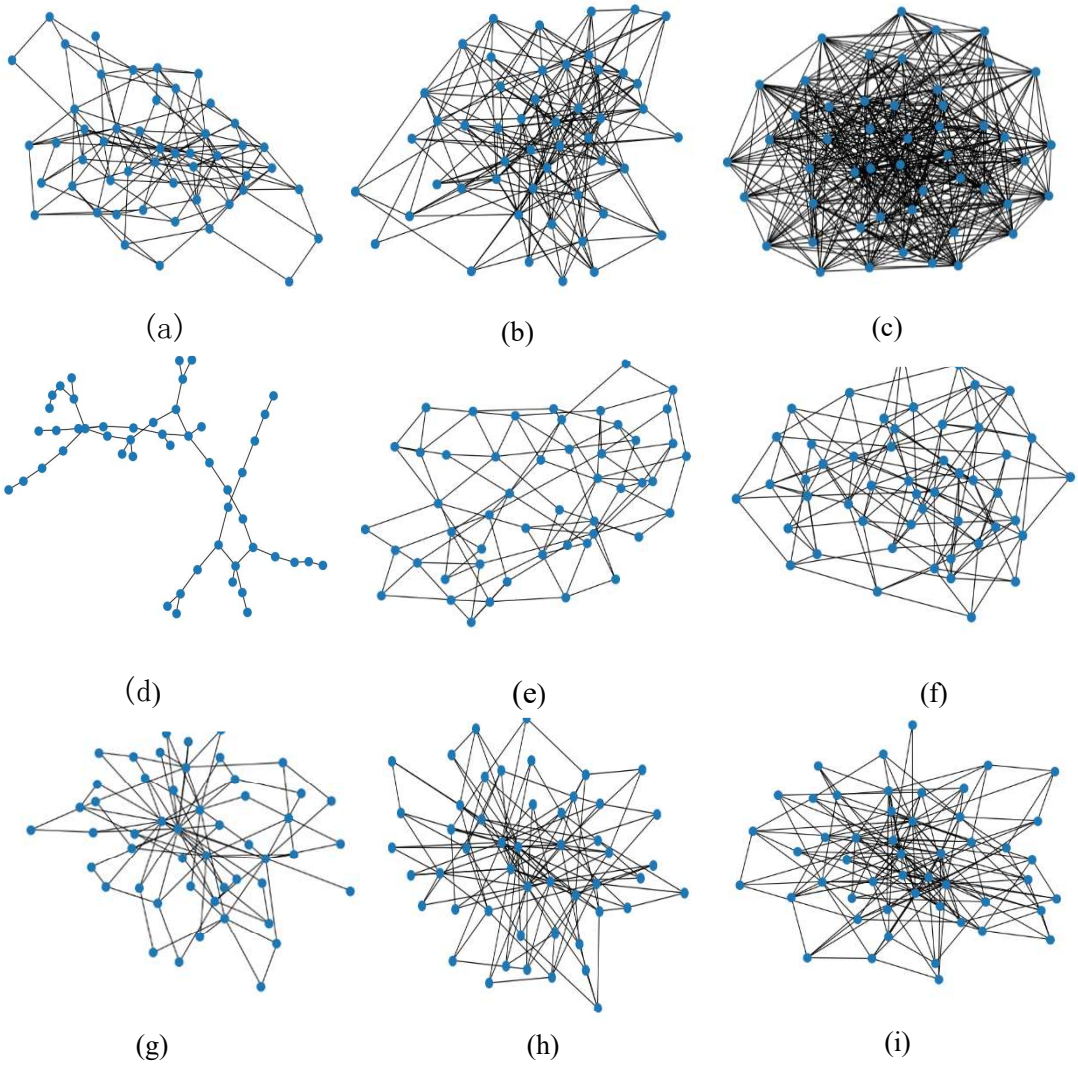


图 5-4 实验拓扑。(a)ER 随机网络,p=0.1;(b) ER 随机网络,p=0.2;(c)ER 随机网络,p=0.4;(d)WS 小世界网络,i=2,p=0.5;(e)WS 小世界网络,i=4,p=0.5;(f)WS 小世界网络,i=6,p=0.5;(g)BA 无标度网络,i=2;(h)BA 无标度网络,i=3;(g)BA 无标度网络,i=4。

5.3.2 仿真方法介绍

本仿真实验共分为三个模块，对应为离线探测准备，流量文件生成，在线故障探测三个部分。具体总流程如图 5-5 所示。

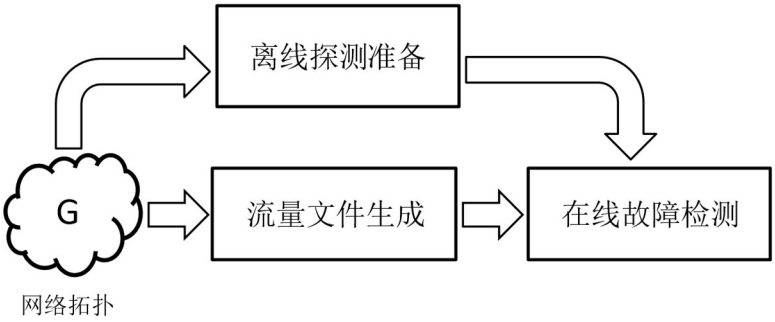


图 5-5 仿真流程

5.3.2.1 离线探测准备

在这一部分，通过输入的网络拓扑、分区数k和待测链路比例r，计算出对应的待测链路集，一轮探测路径以及最小可识别集，并且通过分析可疑区域的所有组合，对对应的二轮探测路径进行离线计算，以供在线故障检测模块使用。路径计算一般是探测中最耗时的部分，离线计算探测路径有效避免了实时计算带来的时间延迟。离线探测准备模块流程和产生的输出如图 5-6 所示：

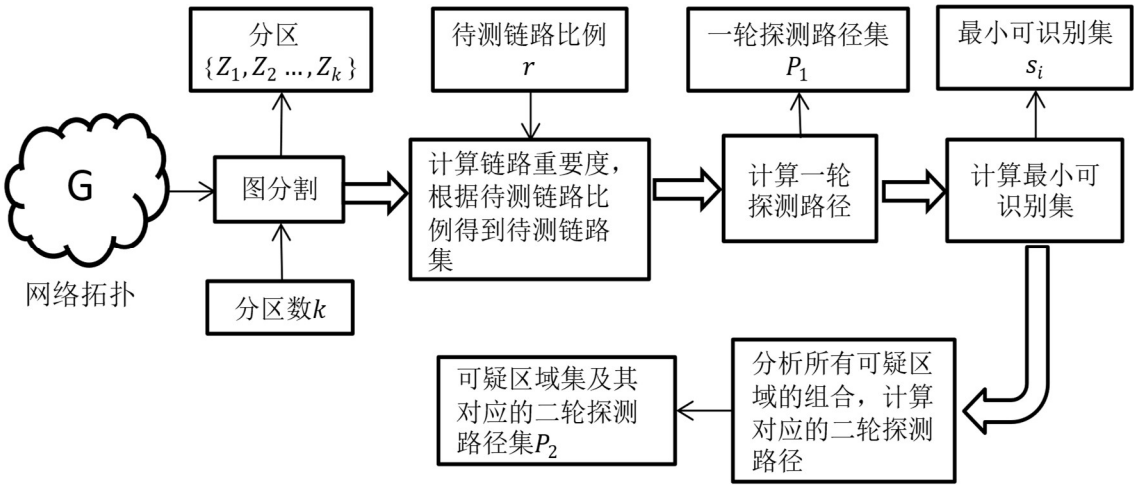


图 5-6 离线探测准备

5.3.2.2 流量文件生成

流量数据使用 python 的 random 包生成，流量的大小服从 Pareto 分布，对每个拓扑生成了 UR，HP，TOR 三种流量模式的流量，每种生成 1000 组，对应 1000 个测量周期，生成的流量数据存于对应的.txt 文件中。每条流的格式如下：

$$(src,\ dst):\ bandwidth \tag{5-1}$$

其中src表示源端，dst表示目的端，$bandwidth$表示该流的带宽大小，文件中的一行包含一个测量周期的所有流的情况。对于确定的拓扑和流量模式，其流量文件是相同的，产生的故障情况也是一致的，有利于其在不同仿真输入下的代价比较。流量文件生成过程如图 5-7 所示

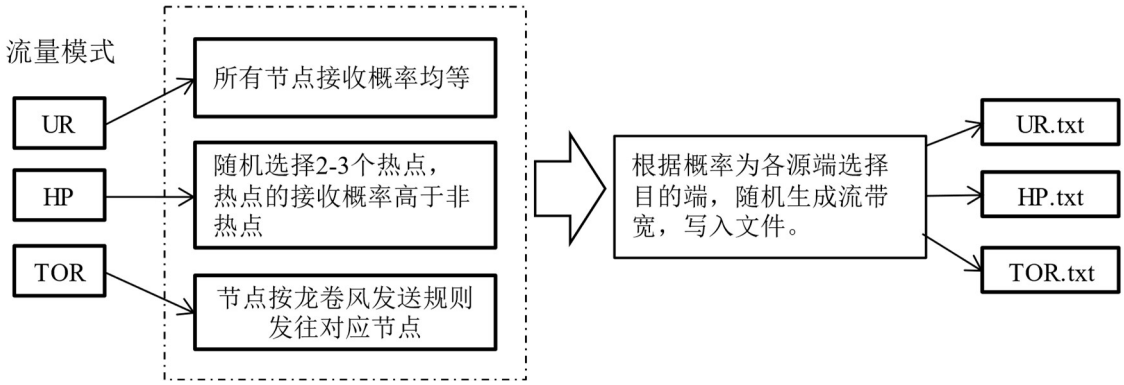


图 5-7 流量文件生成

5.3.2.3 在线故障检测

有了上两节的准备，在在每个测量周期中（对应流量文件的一行），对于每一条流，考虑多路径情况，对于源端和目的端之间的最短路，采取轮询的方式选择传输路径。由于每个测量周期中每条流的源端、目的端和流量大小都是随机生成的，因此故障的存在与否，故障发生的位置等输入信息都是不确定的，需要实时计算，

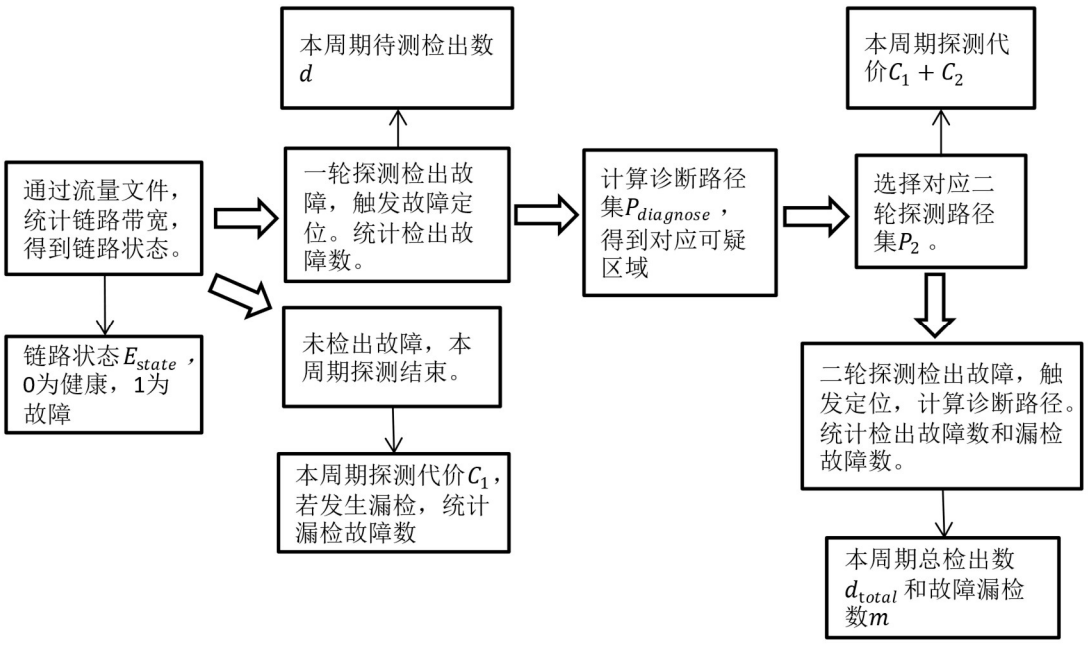


图 5-8 一个测量周期中的在线故障检测

因此称该模块为在线故障定位。在一个测量周期中，首先通过流量文件计算得到所有链路状态，然后得到的一轮探测路径进行对比，一旦探测路径覆盖到故障链路，视为检出故障，随即触发故障定位以及二轮探测。一个测量周期的在线故障检测流程如图 5-8 所示。

5.3.3 不同待测链路比例下的仿真结果分析

根据 4.3.1 节关于测量代价的讨论，本节中的测量代价由式 4-4 给出，统计的是探测路径中经过的链路总数。本文研究的是端到端的测量，即探针站只能设置在端系统上，故不考虑探测站的选址和设置代价。

由前文所述，本文提出的探测方法共分为两轮探测：

第一轮探测的测量路径集P_1是根据链路的重要度排序按比例选出的待测链路集生成的，由于第一轮测量是周期性的测量，第一轮产生的测量代价C_1在待测路径生成后是不变的,令 C_e为沿链路e注入一个探测流的代价，一轮测量代价如式(5-2)。

$$C_1 = \sum_{e \in P_1} C_e \tag{5-2}$$

第二轮探测的进行与否是由第一轮路径的测量结果决定的，并且由于故障发生位置不同，由最小故障集导出的可疑区域也会发生变化，所以第二轮测量的探测路径集P_2和测量代价C_2都是无法预知的。显然，第二轮测量的代价和故障发生的位置和数量息息相关，故障越多，范围越广，其代价应该越大。二轮测量代价如式(5-3)。

$$C_2 = \sum_{e \in P_2} C_e \tag{5-3}$$

C_i表示在第i个测量周期产生的总测量代价，即$C_i = C_{1i} + C_{2i}$，当第一轮测量路径确定后，C_1不变。令$\bar{C}$表示多次测量的平均测量代价，有：

$$\bar{C} = \frac{\sum_{i=1}^{n} C_i}{n} = C_1 + \frac{\sum_{i=1}^{n} C_{2i}}{n} \tag{5-4}$$

在本仿真中，总的故障检出率FDR计算如下式，d_{total}为每个周期中故障的总检出数，m为每个周期中的漏检数：

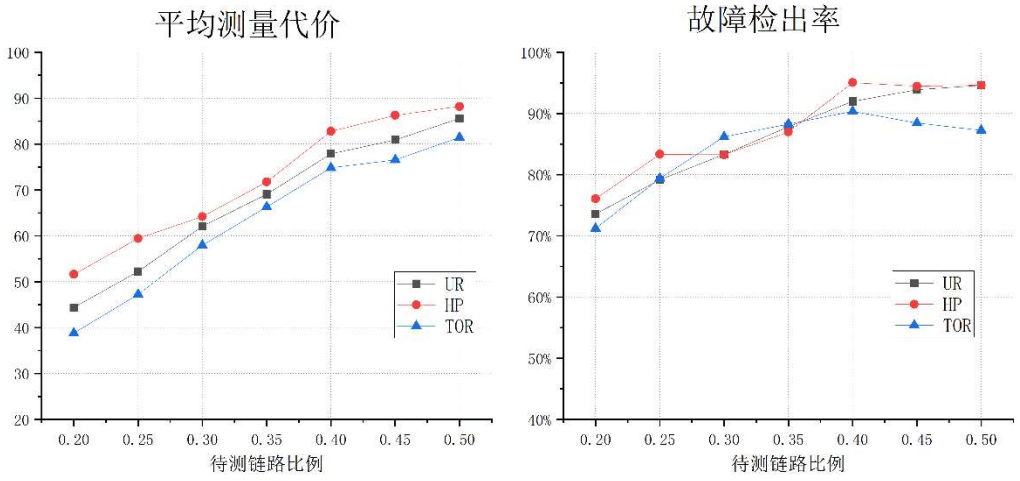
$$FDR = \frac{\sum d_{total}}{\sum (d_{total} + m)} \times 100\% \tag{5-5}$$

上式中分母为所有 1000 个周期的总故障数之和，分子为每个周期的总检出数之和。

- 仿真一：ER 随机网络在不同待测链路比例和不同流量模式下的仿真结果。

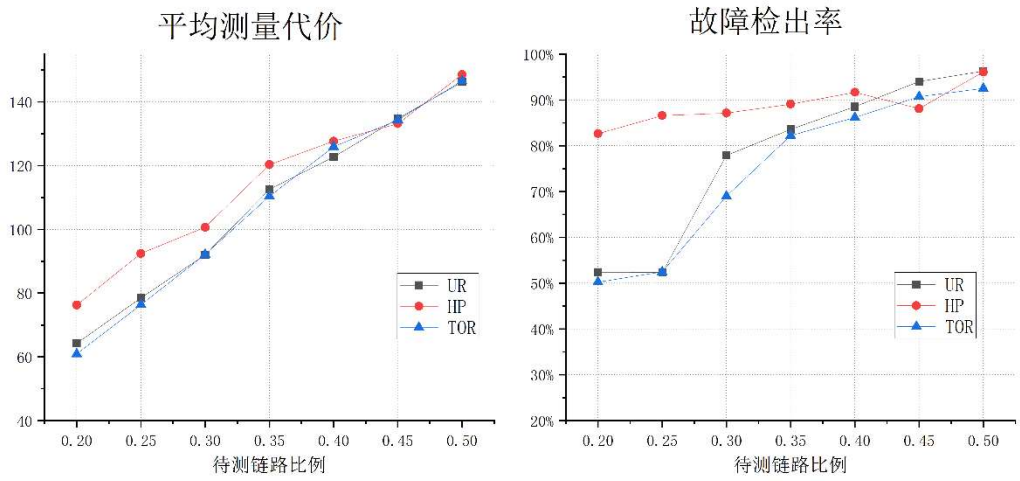
本实验所用得到 ER 随机网络拓扑如图 5-4(a)-(c)所示，仿真结果如图 5-9 至 5-

11 所示，可以看出随着p值的增加，节点的度和拓扑的链路数都随之增加。由于 ER 网络中节点的度都相对均匀，链路分布也相对均匀，平均测量代价和待测链路比例呈现出正相关关系，说明在第一轮测量产生的代价对总代价影响较大，在图 5-11(a) 中尤其明显。在图 5-9 和 5-10 中可以看出相较于 UR 和 TOR，在 HP 流量模式下的测量代价相对较大。这是可能是由于热点的存在使热点周围的链路更易发生故障，这些链路重要性可能并不高，不能在第一轮测量中被覆盖，通常被第二轮测量覆盖和检出，所以在测量代价和故障检出率中 HP 模式下的测量代价会相对更



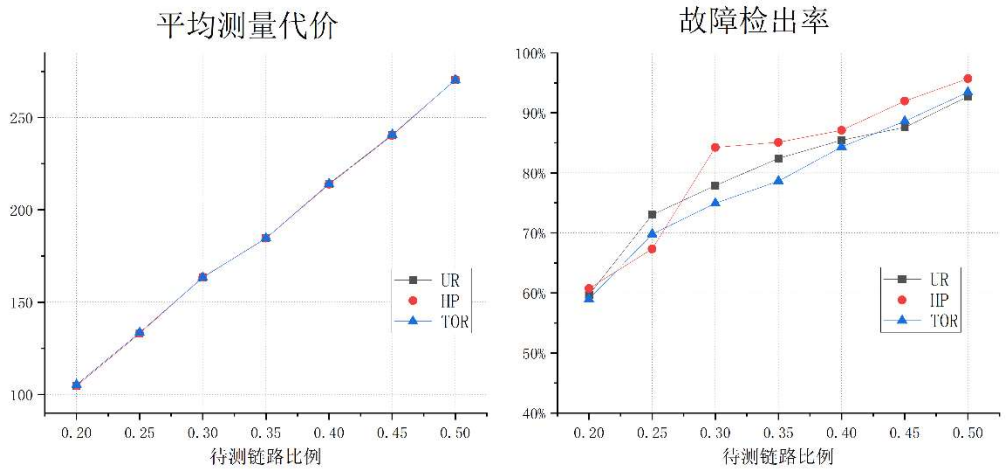
(a) (b)

图 5-9 N=50，p=0.1，$|E|$=148 的 ER 随机网络在不同待测链路比例和不同流量模式下的仿真结果



(a) (b)

图 5-10 N=50，p=0.2，$|E|$=221 的 ER 随机网络在不同待测链路比例和不同流量模式下的仿真结果



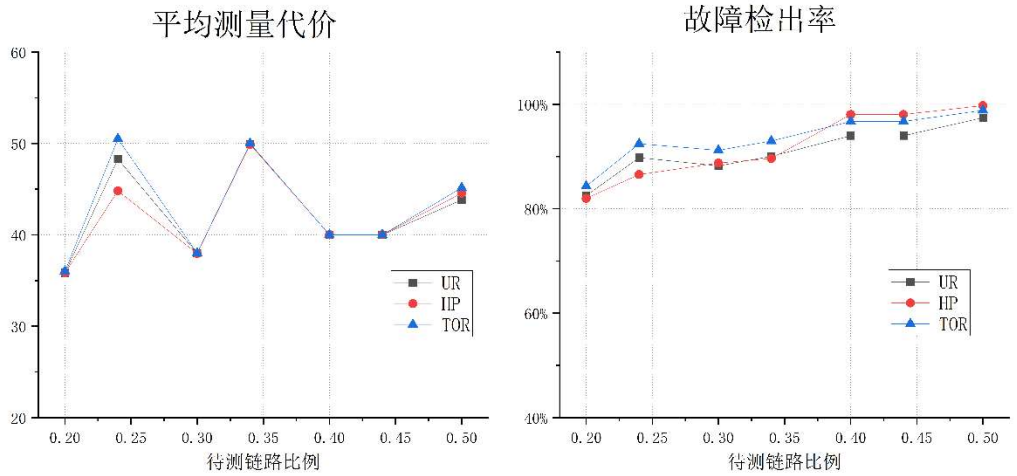
(a) (b)

图 5-11 N=50，p=0.4，$|E|$=480 的 ER 随机网络在不同待测链路比例和不同流量模式下的仿真结果

大。同样，在相同的待测链路比例下，同一拓扑的第一轮测量路径也是相同的，故障检出率的不同由第二轮测量和故障发生的位置共同决定。通过观察发现，在 HP 模式下故障检出率也会相对变大，在图 5-11 中这种变化相当明显，原因可能是在较小的待测路径比例中已经覆盖到了由于 HP 流量的不均衡性所产生的瓶颈链路。

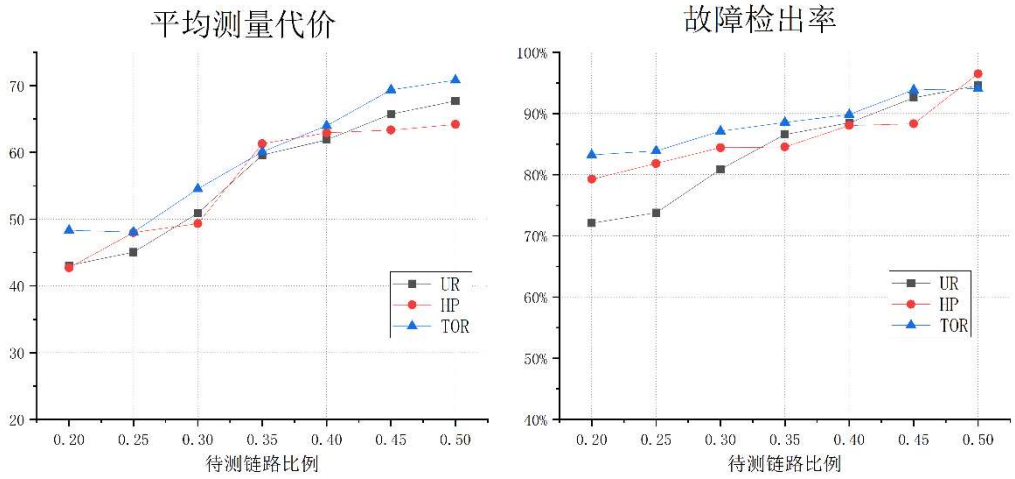
- 仿真二：WS 小世界网络在不同待测链路比例和不同流量模式下的仿真结果

本仿真所使用的 WS 小世界网络拓扑如图 5-4 (d)-(f)所示，结果见图 5-12 至图 5-14。可以看出图(d)具有所有实验拓扑中最少的链路数，端到端之间的路径条数少，流量集中于某些链路中，这些链路具有显而易见的瓶颈特性，正因如此其具



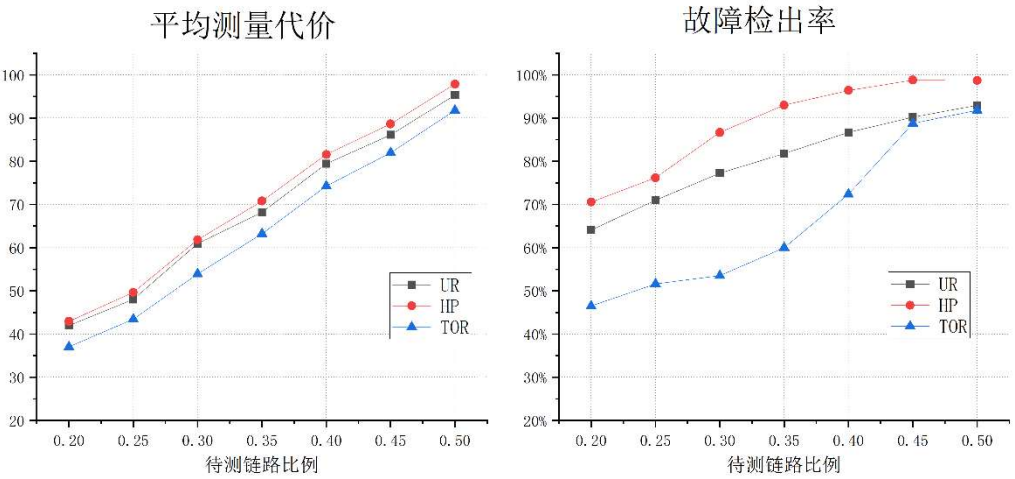
（a） （b）

图 5-12 N=50，i=2，p=0.5，$|E|$=50 的 WS 网络在不同待测链路比例和不同流量模式下的仿真结果



（a）　　（b）

图 5-13 N=50，i=4，p=0.5，$|E|$=100 的 WS 网络在不同待测链路比例和不同流量模式下的仿真结果



（a）　　（b）

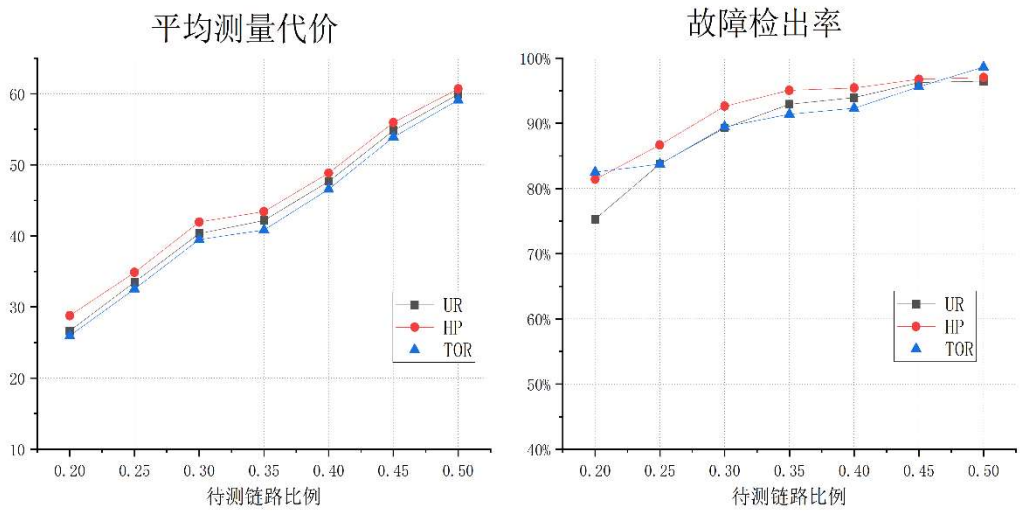
图 5-14 N=50，i=6，p=0.5，|E|=150 的 WS 网络在不同待测链路比例和不同流量模式下的仿真结果。

有所有实验中最多的故障数，可以看出它的平均测量代价不同于其他拓扑，在待测链路比例r=0.3 和 0.4 时都呈现下降；并由于故障链路多发于瓶颈链路，而瓶颈链路又具有较高的传输重要度，即使r很小，大部分瓶颈链路也已被一轮测量覆盖，所以检出率提升不大，甚至在r=0.3 时还出现了小幅度下降。综合来看，在对于图 5-4（d）的 WS 小世界网络来说，最佳的待测链路比例r应在 0.4 左右。对于图 5-4（f）来说，由于链路数最多，发生故障的几率相对较小，平均测量代价呈现出和 ER 网络相似的近似正比的关系，而图 5-4（e）的变化介于两者之间。而由于 TOR

流量相对不均衡，其故障检出率在不同拓扑中变化相对较大，但总体都是随着待测链路比例的增大而增大。

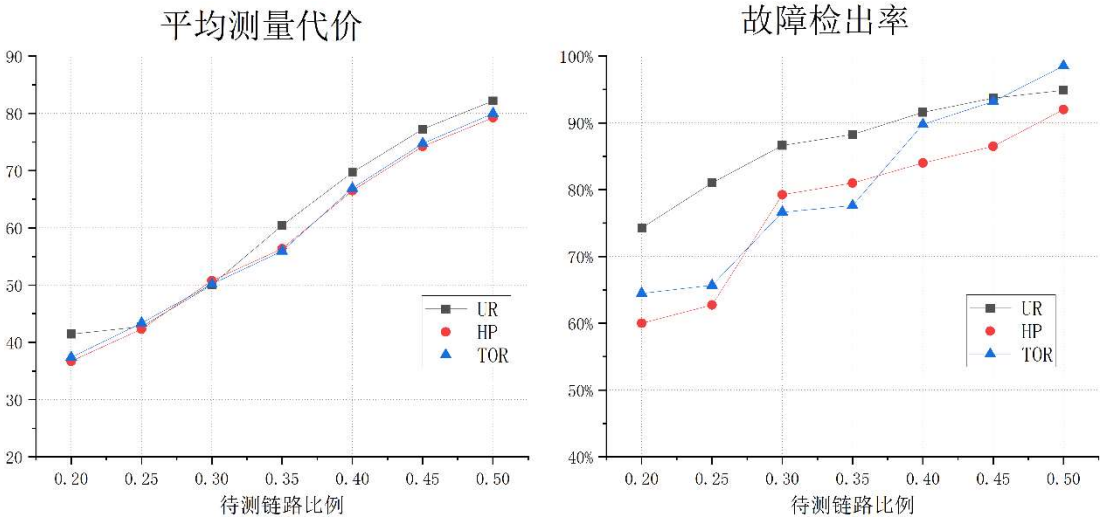
● 仿真三：BA 无标度网络在不同待测链路比例和不同流量模式下的仿真结果。

仿真所使用的 BA 网络拓扑如图 5-4 (g)-(i)所示，结果见图 5-15 至 5-17。可以



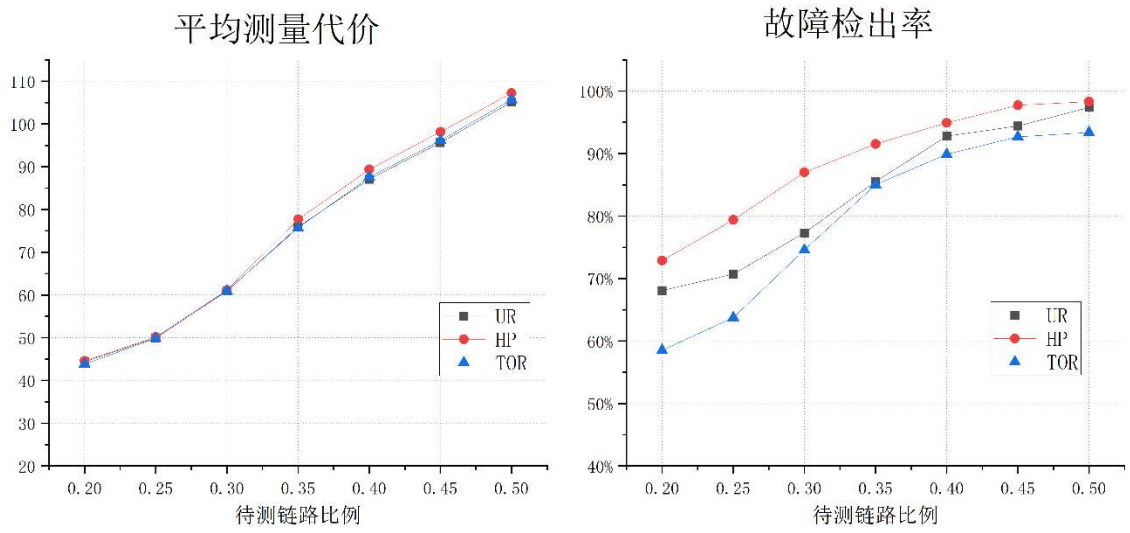
（a） （b）

图 5-15 $|N|$=50，i=2，$|E|$=96 的 BA 网络在不同待测链路比例和不同流量模式下的仿真结果。



（a） （b）

图 5-16 $|N|$=50，i=3，$|E|$=141 的 BA 网络在不同待测链路比例和不同流量模式下的仿真结果。



（a） （b）

图 5-17| N|=50，i=4，|E|=184 的 BA 网络在不同待测链路比例和不同流量模式下的仿真结果。

看出测量代价随待测链路比例的增加呈现出增大的趋势。故障检出率总体也呈现随待测链路比例的增加而增大的趋势，并且增大程度慢慢递减，在 UR 流量模式下尤为明显。HP 模式和 TOR 模式和 UR 模式的区别，主要是由于流量的空间不均衡性带来的。

综合以上三个仿真，在拓扑链路数足够多，端到端之间路径数足够多的情况下，平均测量代价和链路测量额比例呈现出近似正比的关系。而在路径相对单一，瓶颈链路明显的拓扑中，如图 5-4 (d)中的拓扑，链路代价和待测链路比例呈现出波动。主要原因可能是在多路径条件下网络故障发生更少，分布更稀疏，所以发生二轮测量的可能性更低，第一轮测量的测量代价占总测量代价的比例更大。而待测链路数和一轮测量代价C_1近似呈正比关系，即$C_1 = kr|E|$，可以得到：

$$\bar{C} = kr|E| + \frac{\sum_{i=1}^{n} C_{2i}}{n} \tag{5-6}$$

在网络设计合理的情况下，故障是稀疏的。C_2在一轮测量没有故障检出的情况下通常是 0。并且在|E|越大的拓扑中，由于其端到端路径数更多，其中发生故障的概率也会相应减小，在这样的拓扑中，$kr|E|$总是大于$(\sum_{i=1}^{n} C_{2i})/n$，平均测量代价呈现出近似正比的关系。尤其是在流量均衡分布的 UR 模式下，图 5-11（a）、5-14（a），5-17(a)体现得很明显。在 HP 和 TOR 模式下也存在类似趋势，但由于流量的不均衡分布，和 UR 模式相比会存在一些波动。

本测量方法和全覆盖方法的代价对比见图 5-18，对于故障检出率，不同拓扑和流量模式的总体趋势都是随着待测链路比例的提高而增大，只是在增幅存在区别。但总的来说，在$r \geq 0.3$时，基本都能检出 80%以上的故障；在$r \geq 0.45$时，一般能检出 90%以上的故障。图 5-18 对比了在r=0.45 时不同拓扑的平均测量代价和全覆盖测量代价的对比。

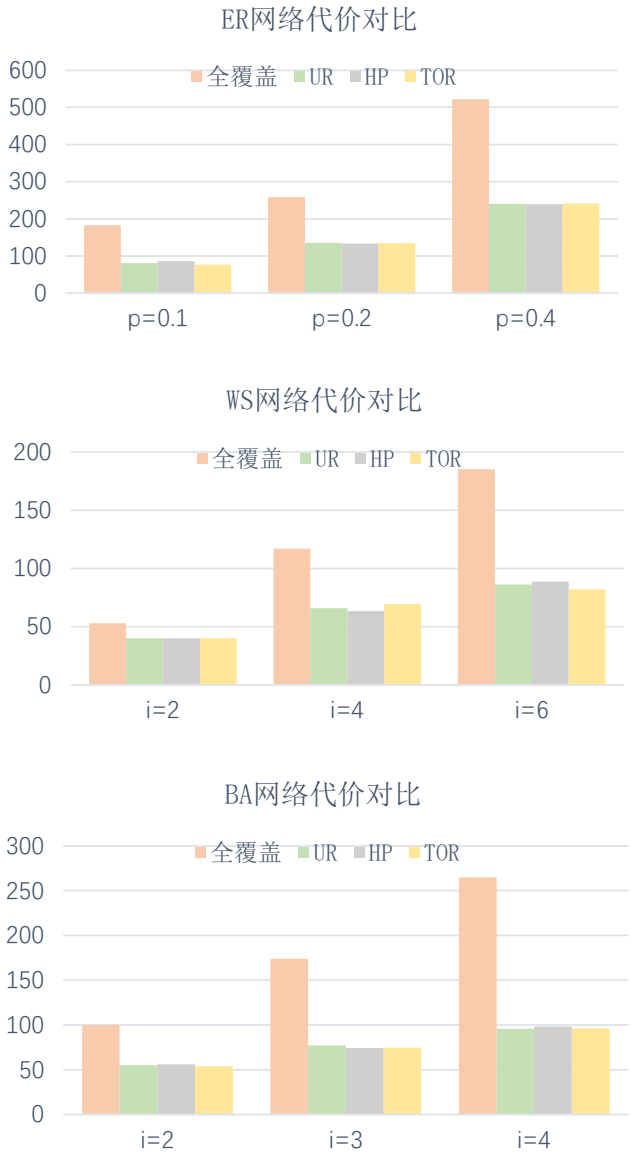


图 5-18 全覆盖测量代价和 r=0.45 时的本测量方法的测量代价对比

可以看出除了在 i=2 的 WS 网络中，大多数拓扑中本方案的测量代价在全覆盖的一半左右，并且拓扑越复杂差距越明显。显然，在周期性的测量中，累积的代价差距将越发可观。以上实验说明在多路径条件下本文提出的故障测量方法能够

用较小的代价测得接近 90%的故障。

5.3.4 考虑额外链路信息的待测路径选择

在 5.3.2 节的仿真中，可以观察到，由于受到不均衡的流量模式的影响，故障检出率可能呈现出某些不规律的变化，这对本方案的适用性是不利的。可以看出在很多情况下，在r=0.3 甚至更小的时候已经取得了不错的测量结果,但考虑到不规律的情况，在 5.3.2 中给出能够保证故障检出率的r是 0.45。可以合理猜测，如果能够将这种不规律性以某种方式量化并引入重要度排序中，可以在保证高检出率的同时降低r的取值。

网络中的故障服从著名的 Pareto 分布，即 80%的故障发生在 20%的链路上。在之前的重要度评价中，我们重点考虑了基于链路的传输重要性，但链路的传输同时受到流量空间分布的影响，对抗性的流量模式会影响链路的状态。能否引入额外的链路信息，使本方法能够以更低的代价得到更可靠的测量结果呢？

链路利用率是一个能够反映链路带宽和流量情况的链路信息。在 SDN 环境下能够很容易的通过控制器和交换机的通信得到该信息，并且不向拓扑注入额外流量。在这一节中，考虑将链路利用率引入得到的重要度，在每一次的测量中重新选择待测链路并且计算测量路径。新的重要度计算方法如下：

1） 将得到的重要度$\{t_1,\dots,t_{|E|}\}$进行归一化处理得到$\{t'_1,\dots,t'_{|E|}\}$

2）获取每条链路的链路利用率$\{u_i,\dots,u_{|E|}\}$

3）令$t'_i = t_i^{'} \times u_i$得到新的$\{t'_1,\dots,t'_{|E|}\}$

为了区别，将新的重要度称为利用率重要度。令r=0.2，使用 5.3.2 节中的流量文件，用改进的重要度评价方法进行仿真，实验结果如图 5-19 至图 5-21。通过对结果的观察，可以发现，使用利用率重要度筛选出的待测链路集能够将测量代价进一步降低，同时具有很高的故障检出率。这说明利用率重要度相比于原来的重要度更能反映链路的瓶颈程度，能更好的覆盖到故障链路。

值得注意的是，除了图 5-16 中 i=2 的 WS 网络，其余网络中待测链路的故障检出率和总故障检出率的差别极小。这意味着二轮测量几乎没有检测出故障，绝大部分故障都被一轮测量覆盖。这意味着使用利用率重要度评价方法可以取消二轮测量以进一步减小测量代价。并且在这些网络中的故障检出率都非常接近全覆盖的方法，也就是有接近 100%的检出率。这是因为链路利用率和链路故障，尤其是拥塞和丢包，具有很强的相关性。将链路利用率加入重要度评价能够很好的反应链路的状态。

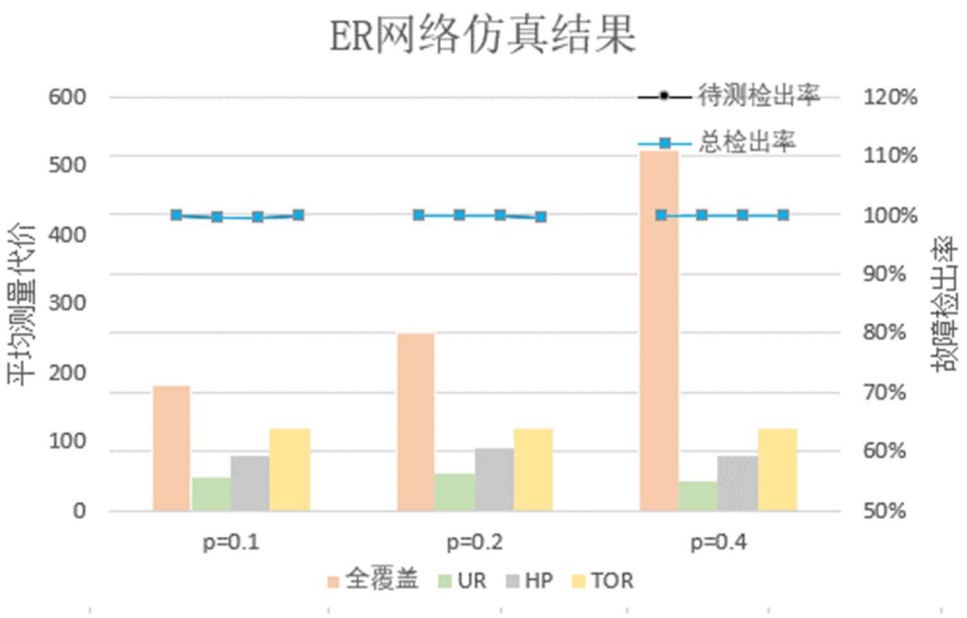


图 5-19 r=0.2，p=0.1、0.2、0.4 的 ER 随机网络的利用率重要度仿真结果

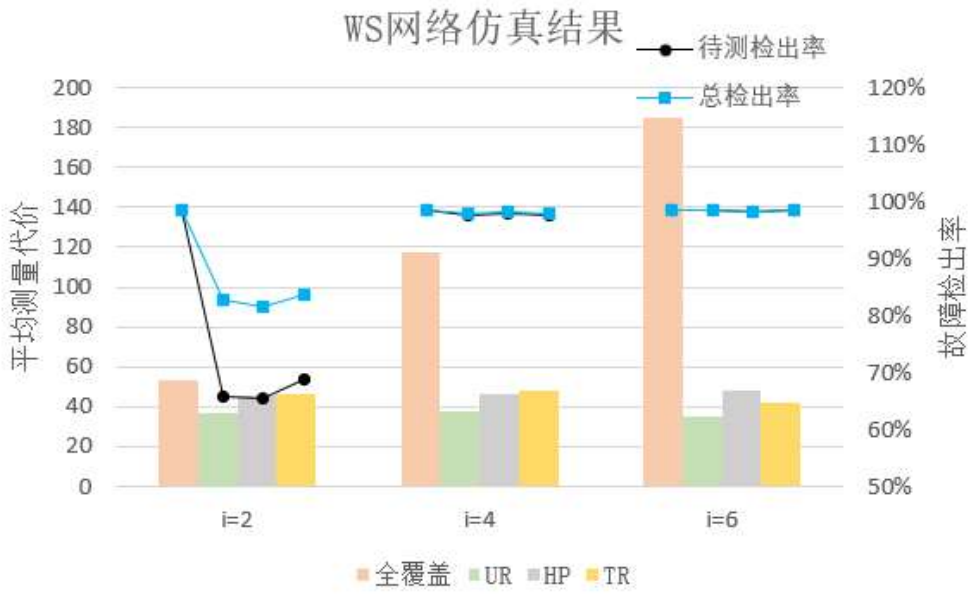


图 5-20 r=0.2，i=2、4、6 的 WS 随机网络的利用率重要度仿真结果

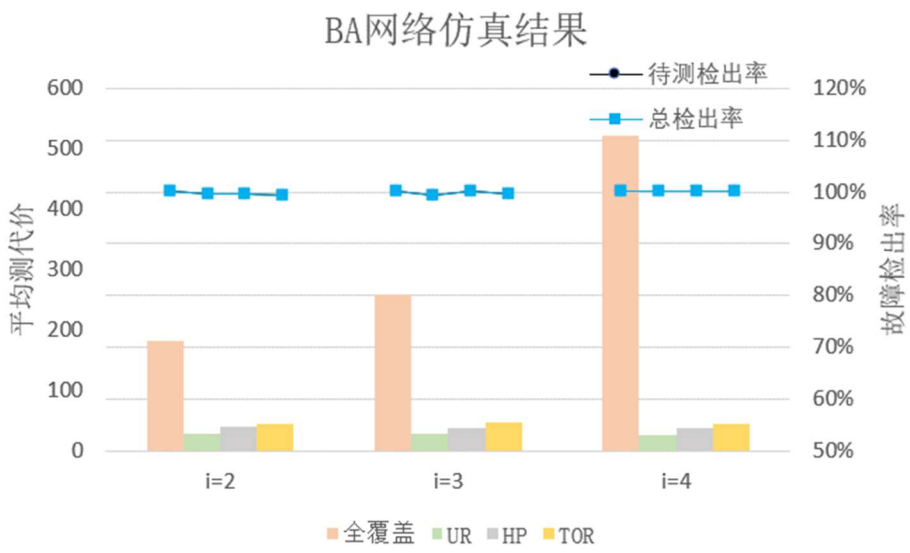


图 5-21 r=0.2，i=2、3，4 的 BA 无标度网络的利用率重要度仿真结果

对于图 5-20 中i=2 的 WS 网络，由于该网络链路较少，端与端之间的路径条数也很少，链路成为瓶颈的概率大，在该网络中的故障发生率会更高。在这样的情况下，r=0.2 的选择比例不足以覆盖到所有故障链路。也是因为这个原因，在对抗性流量模式下故障检出率会更高，链路利用率能很好的反映流量的不均衡分布。

虽然利用率重要度的测量代价很低并且效果很好，但其在一般的网络中获得利用率的难度大，并且由于链路利用率是一个动态变量，需要在每个测量周期实时取得，因此重要度的计算、待测链路的筛选和探测路径的计算都需要在测量周期中在线进行，这会导致不可忽略的延迟，并且越大越复杂的网络需要收集的信息越多，产生的延迟也会更大。而原本的方法所有计算都是离线的，在网络拓扑不发生变化的情况下，只需要在测量开始时进行一次，产生的延迟会相对小很多，并且不需要存储额外的链路信息。两种方式各有利弊，需要结合具体使用场景进行调整和选择。

5.4 本章小结

本章首先介绍了仿真所需的拓扑模型和流量模式，接着展示了生成的具体拓扑，并对仿真涉及的各个模块和流程进行了介绍。然后对评价测量代价进行了分析，并用不同待测比例结合不同的拓扑和流量模式进行了仿真对比实验，分析了结果，并和全覆盖的测量代价进行了对比，证明了本文测量方案能够小的测量代价下测得大部分故障。最后向重要度添加了额外链路信息——链路利用率，以进一步减小测量代价，实验证明了利用率重要度能在很低的测量代价下取得好的效果。

第六章　总结和展望

6.1 本文工作总结

随着网络规模的不断扩大，为了提供可靠的服务，对网络中的故障进行周期性的检测和定位是至关重要的。主动测量方法由于其灵活性被广泛运用于网络的故障测量中，但主动测量是侵入式的测量，会向网络中注入额外的探测流量，可能会影响网络的正常功能。

为了降低端到端多路径的链路级故障的主动测量的带宽代价，本文提出了一套基于图分割和链路重要度排序的网络故障检测方法，该方法共分为两轮测量。本文的主要工作如下：

（1） 介绍了网络测量的相关概念，回顾并分析了主要的节点和链路重要度评价算法，以及图分区方法。并对网络故障检测和定位进行了研究。这些研究为设计和提出本文的测量方案打下了技术基础。

（2） 探讨了分区测量相比于全覆盖方案的代价优势；分析并选择了适合的图分区算法；为了更适应本文的测量需求，提出了基于分区链路传输重要度评价方法，相比于对比方法可以减少需要统计的路径数，同时提高域间链路的被覆盖可能性，以减少漏检。利用重要度评价方法，筛选出了待测链路集，并基于此设计了探针选择算法，选出了第一轮测量的探测路径。

（3） 利用第一轮测量探针返回的结果，对结果中得到的故障路径进行了定位，首先探讨了链路故障和路径故障的关系，提出了最小可识别集的概念，设计了故障定位算法。定位到的最小可识别集被称为最小故障集，最小故障集对应的区域为可疑区域。可疑区域即为存在故障的区域。为了对一轮测量未覆盖的链路进行进一步测量，提出了针对可疑区域的全覆盖测量方法。

（4） 根据具体场景，设计了仿真实验，对不同的拓扑在不同流量模式下使用不同比例待测链路的平均测量代价和故障检出率进行了实验。实验证明本文的测量方法能够在较小的测量代价下实现较高的故障检出率。为了更好的适应各种流量场景，对前文重要度评价进行了改进，考虑了额外的链路信息，链路利用率。实验证明，新的重要度能够在更小的代价下实现更加可靠的重要度检测，但在即时性上会有所下降。

6.2 后续工作展望

本文提出了一种基于分区和链路重要度的多步网络测量方法，并对其进行了数

值仿真。但由于时间的不足和技术条件的限制，还存在很多可以改进的问题，可以作为后续工作的展望。

（1） 只进行了数值仿真验证了测量方法的基本正确性，未在真实网络环境下进行实际的发包测量。在后续的工作中，应该将测量部署在实际网络中，以研究真实情况下的性能。

（2） 对于检测出故障的下一步处理没有进行的研究和探讨。后续的工作应该加入这方面的研究，并且在实际拓扑中进行实现。

（3） 本文采用的图分割方法，是基于现有的性能较好的图分割算法而得到的。在本研究中假设网络拓扑是不变的，仅考虑了基于不变拓扑的离线分割。在后续的工作中，可以考虑利用网络中的其他链路和节点信息，并且考虑网络拓扑的动态性自适应地划分拓扑。

致　谢

在电子科大的这三年，我认识了很多人，也受到了大家的很多帮助。

首先当然要感谢我的导师许都教授。许老师个人特点非常鲜明，他不是事无巨细的老师，给予每个学生充分的自由，却又关心大家的前途发展。在我实习和求职的过程中都非常的包容和支持我。由于疫情和实习的耽误，我的毕业设计进展并不是很顺利，但每当我陷入瓶颈，许老师总能够从他渊博的知识中带给我新的启发。有段时间我一直没有想出方案，一度非常焦虑，许老师总是会说：“哪会有那么容易呢？”许老师的从容淡定让我不再急躁，沉下心来继续思考。许老师也是一个非常有趣的老师，和他交流的时候总会听见他爽朗的笑声。再次衷心感谢这三年间恩师的关心、教导和帮助，祝老师身体健康，笑口常开。

其次我要感谢我的室友和 kb310 的同学们，感谢我的室友顾静玲对我生活和学习上的帮助，很感谢这三年的陪伴。感谢 310 的同学们帮我思考讨论问题，在我消沉的时候安慰我，一起加班到深夜互相打气。我不是一个非常外向的人，但大家的热心和友好让我融入了这个集体，感受到了互相帮助的温暖，祝愿大家学业顺利、前途似锦。

还要感谢我的家人，尤其是我的父亲，我的父亲是最支持我的人，他总是无条件的信任我、鼓励我、安慰我。以后我也想成为父亲的依靠，希望爸爸永远年轻、身体健康。

参考文献

[1] H. Gao, L. Zhao, H. Wang, et al. XShot: Light-weight Link Failure Localization using Crossed Probing Cycles in SDN[C].49th International Conference on Parallel Processing-ICPP. 2020: 1-11.

[2] Y. Chen, W. Zhou, Z. M. Duan, et al. Congested link loss rate range inference algorithm in IP network[J]. Ruan Jian Xue Bao/Journal of Software, 2017,28(5):1296-1314 (in Chinese). http://www.jos.org.cn/1000-9825/5148.htm

[3] K. Andreev, H. Racke. Balanced graph partitioning[J]. Theory of Computing Systems, 2006, 39(6): 929-939.

[4] B. W. Kernighan, S. Lin. An efficient heuristic procedure for partitioning graphs[J]. The Bell system technical journal, 1970, 49(2): 291-307.

[5] G. Sabidussi. The centrality index of a graph[J]. Psychometrika, 1966, 31(4): 581-603.

[6] L. C. Freeman. A set of measures of centrality based on betweenness[J]. Sociometry, 1977: 35-41.

[7] S. Brin, L. Page. The anatomy of a large-scale hypertextual web search engine[J]. Computer networks and ISDN systems, 1998, 30(1-7): 107-117.

[8] W. Liu, L. Lü. Link prediction based on local random walk[J]. EPL (Europhysics Letters), 2010, 89(5): 58007.

[9] 任晓龙, 吕琳媛. 网络重要节点排序方法综述[J]. 科学通报, 2014, 59: 1175－1197

[10] X. Cheng, F. X. Ren, H. W. Shen, et al. Bridgeness: a local index on edge significance in maintaining global connectivity[J]. Journal of Statistical Mechanics: Theory and Experiment, 2010, 2010(10): P10011.

[11] 蒋林承. 复杂网络节点和边重要性度量以及多源信息传播问题研究[D].国防科技大学,2018.

[12] H.Corley, Y.S.David,Most vital links and nodes in weighted networks[J],Operations Research Letters,1982,1(4):157-160.

[13] 陈勇，胡爱群，胡啸.通信网中节点重要性的评价方法[J]，通信学报，2004，25（8）：129-134.

[14] M. Kitsak, L. K Gallos, S. Havlin, et al. Identification of influential spreaders in complex networks[J]. Nature physics, 2010, 6(11): 888-893.

[15] D. B. Chen, L. Lü, M. S. Shang, et al. Identifying influential nodes in complex networks[J]. Physica A, 2012, 391(4): 1777–1787

[16] J. M. Kleinberg. Authoritative sources in a hyperlinked environment[J]. Journal of the ACM (JACM), 1999, 46(5): 604-632.

[17] S. Chakrabarti, B. Dom, P. Raghavan, et al. Automatic resource compilation by analyzing hyperlink structure and associated text[J]. Computer networks and ISDN systems, 1998, 30(1-7): 65-74.

[18] R. Lempel, S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect[J]. Computer Networks, 2000, 33(1-6): 387-401.

[19] M. S. Granovetter. The strength of weak ties[J]. American journal of sociology, 1973, 78(6): 1360-1380.

[20] F. Radicchi, C. Castellano, F. Cecconi, et al. Defining and identifying communities in networks[J]. Proceedings of the national academy of sciences, 2004, 101(9): 2658-2663.

[21] E. Gilbert, K. Karahalios. Predicting tie strength with social media[C].Proceedings of the SIGCHI conference on human factors in computing systems. 2009: 211-220.

[22] J-P. Onnela,J. Saramaki,J. Hyvonen,G. Szabo , et al. Barabasi, Structure and tie strengths in mobile communication networks[J], Proceedings of the National academy of Sciences,2007,104(18):7332-7336.

[23] M. Girvan., M. E. Newman.,Community structure in socialand biological networks[J], Proceedings of the Nati onal academy of Sciences,2002,99(12):7821-7826.

[24] P. De Meo, E. Ferrara, G. Fiumara, et al. A novel measure of edge centraliy in social networks[J], Knowledge-based systems,2012,30 136-150.

[25] 姜禹,胡爱群,何明.基于网络传输特性的链路重要性评价方法[J].中国工程科学,2009,11(09):64-67+73.

[26] A. Moffat, T. Takaoka. An all pairs shortest path algorithm with expected running time O (n 2logn)[C].26th Annual Symposium on Foundations of Computer Science (sfcs 1985). IEEE, 1985: 101-105.

[27] 郑丽丽.图划分算法综述[J].科技信息,2014(04):145-148.

[28] C. M. Fiduccia, R. M. Mattheyses. A linear-time heuristic for improving network partitions[C].19th design automation conference. IEEE, 1982: 175-181.

[29] G. Karypis, V. Kumar. Multilevel k-way partitioning scheme for irregular graphs[J]. Journal of Parallel and Distributed Computing January 1998; 48(1):96–129.

[30] L. Hyafil, R. L. Rivest. Graph partitioning and constructing optimal decision trees are polynomial complete problems[M]. IRIA. Laboratoire de Recherche en Informatique et Automatique, 1973.

[31] K. Andreev, H. Racke. Balanced graph partitioning[J]. Theory of Computing Systems, 2006, 39(6): 929-939.

[32] M. Holtgrewe, P. Sanders, C. Schulz. Engineering a scalable high quality graph partitioner[C].2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS). IEEE, 2010: 1-12.

[33] R. Diekmann, B. Monien, R. Preis. Using helpful sets to improve graph bisections[J]. Interconnection networks and mapping and scheduling parallel computations, 1995, 21: 57-73.

[34] B. Monien, S. Schamberger. Graph partitioning with the party library: Helpful-sets in practice[C].16th Symposium on Computer Architecture and High Performance Computing. IEEE, 2004: 198-205.

[35] W. E. Donath, A. J. Hoffman. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices[J]. IBM Technical Disclosure Bulletin, 1972, 15(3): 938-944.

[36] W. E. Donath, A. J. Hoffman. Lower bounds for the partitioning of graphs[M].Selected Papers Of Alan J Hoffman: With Commentary. 2003: 437-442.

[37] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory[J]. Czechoslovak Mathematical Journal, 1975, 25(4): 619-633.

[38] S. T. Barnard, H. D. Simon. Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems[J]. Concurrency: Practice and experience, 1994, 6(2): 101-117.

[39] B. Hendrickson, R. Leland, Van. Driessche. R. Enhancing data locality by using terminal propagation[C].Proceedings of HICSS-29: 29th Hawaii International Conference on System Sciences. IEEE, 1996, 1: 565-574.

[40] G. Karypis, V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs[J]. SIAM Journal on scientific Computing, 1998, 20(1): 359-392.

[41] A. George, J. W. Liu. Computer solution of large sparse positive definite[M]. Prentice Hall Professional Technical Reference, 1981.

[42] L. R. Ford, D. R. Fulkerson. Maximal Flow through a Network, Canadian Journal of Mathematics[J]. 1956.

[43] H. D. Simon. Partitioning of unstructured problems for parallel processing[J]. Computing systems in engineering, 1991, 2(2-3): 135-148. .

[44] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations[J]. Concurrency: Practice and experience, 1991, 3(5): 457-481.

[45] C Farhat, M Lesoinne. Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics[J]. International Journal for Numerical Methods in Engineering, 1993, 36(5): 745-764.

[46] G. L. Miller, S. H. Teng, Vavasis S A. A unified geometric approach to graph separators, FOCS'01[J]. 1991.

[47] J. R. Gilbert, G. L. Miller, Teng. S. H. Geometric mesh partitioning: Implementation and experiments[J]. SIAM Journal on Scientific Computing, 1998, 19(6): 2091-2110.

[48] J. R. Pilkington, S. B. Baden. Partitioning with spacefilling curves[J]. 1994.

[49] M. Bader. Space-filling curves: an introduction with applications in scientific computing[M]. Springer Science & Business Media, 2012.

[50] S. Schamberger, J. M. Wierum. A Locality Preserving Graph Ordering Approach for Implicit Partitioning: Graph-Filing Curves[C].ISCA PDCS. 2004: 51-57.

[51] S. Kirmani, P. Raghavan. Scalable parallel graph partitioning[C].SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE, 2013: 1-10.

[52] G. Karypis, V. Kumar. Multilevelk-way partitioning scheme for irregular graphs[J]. Journal of Parallel and Distributed computing, 1998, 48(1): 96-129.

[53] B. Hendrickson, R. W. Leland. A Multi-Level Algorithm For Partitioning Graphs[J]. SC, 1995, 95(28): 1-14.

[54] P. Sanders, C. Schulz. Think locally, act globally: Highly balanced graph partitioning[C]. International Symposium on Experimental Algorithms. Springer, Berlin, Heidelberg, 2013: 164-175.

[55] P. Sanders, C. Schulz. High quality graph partitioning[J]. Graph Partitioning and Graph Clustering, 2012, 588(1): 1-17.

[56] R. Diekmann, R. Preis, F. Schlimbach, et al. Shape-optimized mesh partitioning and load balancing for parallel adaptive FEM[J]. Parallel Computing, 2000, 26(12): 1555-1581.

[57] C. H. Walshaw, M. Cross, M. G. Everett. A localized algorithm for optimizing unstructured mesh partitions[J]. The International Journal of Supercomputer Applications and High Performance Computing, 1995, 9(4): 280-295.

[58] L. Guan, Y. Wang, W. Li, et al. Efficient probing method for active diagnosis in large scale network[C].Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013). IEEE, 2013: 198-202.

[59] M. Brodie, I. Rish, S. Ma. Optimizing probe selection for fault localization[J]. Proceedings of Twelfth International Workshop on Distributed Systems, DSOM 2001, Nancy, France, 2001; 88–98.

[60] M. Brodie, I. Rish, S. Ma, et al. Active probing strategies for problem diagnosis in distributed systems[C].IJCAI. 2003, 3: 1337-1338.

[61] I. Rish, M. Brodie, N. Odintsova, et al. Real-time problem determination in distributed systems using active probing[C].2004 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2004, 1: 133-146.

[62] I. Rish, M. Brodie, S. Ma, et al. Adaptive diagnosis in distributed systems[C]. IEEE Transactions on Neural Networks September 2005; 16(5):1088–1109.

[63] A. X. Zheng, I. Rish, A. Beygelzimer. Efficient test selection in active diagnosis via entropy approximation[J]. arXiv preprint arXiv:1207.1418, 2012.

[64] Y. Tang, E. S. Al-Shaer, R. Boutaba. Active integrated fault localization in communication networks[C].2005 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005. IM 2005. IEEE, 2005: 543-556.

[65] M. Natu, A S Sethi. Active probing approach for fault localization in computer networks[C].2006 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services. IEEE, 2006: 25-33.

[66] M. Natu, A. S. Sethi. Probabilistic fault diagnosis using adaptive probing[C].International Workshop on Distributed Systems: Operations and Management. Springer, Berlin, Heidelberg, 2007: 38-49.

[67] M. Natu, A. S. Sethi. Efficient probing techniques for fault diagnosis[C].Second International Conference on Internet Monitoring and Protection (ICIMP 2007). IEEE, 2007: 20-20.

[68] M. Natu, A. S. Sethi, E. L. Lloyd. Efficient probe selection algorithms for fault diagnosis[J]. Telecommunication systems, 2008, 37(1): 109-125.

[69] M. Natu, A. S. Sethi. Application of adaptive probing for fault diagnosis in computer networks[C].NOMS 2008-2008 IEEE Network Operations and Management Symposium. IEEE, 2008: 1055-1060.

[70] L. Lu, Z. Xu, W. Wang, et al. A new fault detection method for computer networks[J]. Reliability Engineering and System Safety June 2013; 114:45–51.

[71] M. Natu, A.S. Sethi. Probe station placement for fault diagnosis[C]. Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM 2007, Washington D.C., 2007; 113–117.

[72] D. Jeswani, M. Natu, R.K. Ghosh. Adaptive monitoring: Application of probing to adapt passive monitoring[J]. Journal of Network and Systems Management 2015; 23(4):950–977.

[73] R. Carmo, J. Hoffmann, V. Willert, et al. Making active-probing-based network intrusion detection in Wireless Multihop Networks practical: A Bayesian inference approach to probe selection[C]. Proceedings of the Thirty Ninth Annual IEEE Conference on Local Computer Networks, Edmonton, Canada, 2014;345–353.

[74] M. S. Garshasbi. Fault localization based on combines active and passive measurements in computer networks by ant colony optimization[J]. Reliability Engineering & System Safety April 2016; 152:205–212.

[75] M. Dorigo, M. Birattari, T. Stutzle. Ant colony optimization[J]. IEEE computational intelligence magazine, 2006, 1(4): 28-39.

[76] A. Dusia, A. S. Sethi. Recent advances in fault localization in computer networks[J]. IEEE Communications Surveys & Tutorials, 2016, 18(4): 3030-3051.

[77] S. Ramanathan, Y. Kanza, B. Krishnamurthy. SDProber: A software defined prober for SDN[C].Proceedings of the Symposium on SDN Research. 2018: 1-7.

[78] E. Salhi, S. Lahoud, B. Cousin. Localization of Single Link-Level Network Anomalies: Problem Formulation and Heuristic[J]. 2013.

[79] 陈宇,周巍,段哲民,钱叶魁,赵鑫.一种 IP 网络拥塞链路丢包率范围推断算法[J].软件学报,2017,28(5):1296-1314.

[80] S. Pan, P. Li, D. Zeng, et al. A ${Q} $-Learning Based Framework for Congested Link Identification[J]. IEEE Internet of Things Journal, 2019, 6(6): 9668-9678.

[81] M. Brodie, I. Rish, S. Ma, et al. Active probing strategies for problem diagnosis in distributed systems[C].IJCAI. 2003, 3: 1337-1338.

[82] E. Salhi. Detection and localization of link-level network anomalies using end-to-end path monitoring[D]. Rennes 1, 2013.

[83] N. Duffield. Simple network performance tomography[C].Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. 2003: 210-215.

[84] N. Handigol, B. Heller, V. Jeyakumar, et al. I know what your packet did last hop: Using packet histories to troubleshoot networks[C].11th USENIX Symposium on Networked Systems Design and Implementation(14). 2014: 71-85.

[85] Y. Zhu, N. Kang, J. Cao, et al. Packet-level telemetry in large datacenter networks[C].Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. 2015: 479-491.

[86] C. Yu, C. Lumezanu, A. Sharma, et al. Software-defined latency monitoring in data center networks[C].International Conference on Passive and Active Network Measurement. Springer, Cham, 2015: 360-372.

[87] N. L. M. Van. Adrichem, C. Doerr, F. A. Kuipers. Opennetmon: Network monitoring in openflow software-defined networks[C].2014 IEEE Network Operations and Management Symposium (NOMS). IEEE, 2014: 1-8.

[88] C. Guo, L. Yuan, D. Xiang, et al. Pingmesh: A large-scale system for data center network latency measurement and analysis[C].Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. 2015: 139-152.

攻读硕士学位期间取得的成果

[1] 2018-2019 学年，电子科技大学学业二等奖学金

[2] 2018-2019 学年，电子科技大学优秀研究生

[3] 2019-2020 学年，电子科技大学学业三等奖学金