

摘要

网格技术的出现被誉为信息技术的第三次浪潮，它通过使用通用的协议与接口将分布在不同地理位置的各类资源协同起来为用户提供服务。近年来随着开放网格服务架构的概念与标准的提出，网格技术进入了高速发展的阶段。网络安全作为网格的核心问题，目前仅采用安全协议与安全通信的方式提供安全保障是不够的。入侵检测作为传统网络中的重要安全保障手段，为网络提供了主动防御的防护机制。在网格环境中使用入侵检测技术，研究适合于网格环境的入侵检测系统，可以有效地弥补现有网络安全技术的不足，是网络安全技术的一个重要发展趋势。

本文通过分析网格环境的安全问题与入侵方式，并针对现有网格中入侵检测系统研究的不足，提出了一种面向网格的入侵检测系统，系统采用模块化设计方法。文中对该系统的各功能模块的设计与实现进行了深入的研究，并针对网格环境特点，结合该系统提出了一个网格中入侵检测系统的负载均衡算法，算法包括网格环境的模型化与基于图论理论的负载均衡算法两部分。通过负载均衡算法调度，系统的各模块在网格环境中协同工作，达到各模块负载程度均衡，系统总负载降低的目的，提高了面向网格的入侵检测系统整体性能。本文实现的面向网格的入侵检测系统具有扩展性强、动态负载均衡、拥有自适应能力的特点，较好的解决了入侵检测系统难于适应网格环境的问题，并在一定程度上弥补了现有网络安全技术的不足。

本系统采用 Web 服务的方式实现，符合了开放网格服务架构的特点。并在“天津市郊区工业网”项目中进行了部署，为该网格环境提供了入侵检测服务。通过实际使用与实验，验证了该系统对网格环境可以进行有效检测，并且具有良好扩展性与负载均衡能力。

关键字：网格 入侵检测 负载均衡算法 Web 服务

Abstract

Grid has been called the third wave of information technology. It serves the users by coordinating various resources from different places via common protocol and interface. With the development of the concept and standard of OGSA over the past few years, Grid technology has advanced rapidly. As the core of Grid, Grid Security provides safety through security protocol and security transaction, which is not as effective as expected. Intrusion detection, a vital means for the security of traditional webs, defends the webs in an active manner. Applied to the Grid, intrusion detection may evolve into one adapted to the Grid. It will be an effective redemption to the defects of present grid security technology. Hopefully, this will be the future of grid security.

This paper is an attempt to analyze the security problems and intrusion ways in Grid. With a view to improving the existing intrusion detection, the author proposes a grid-oriented intrusion detection system which is based on module-based design and gives the design and implementation of all modules in this system. Aimed at the disadvantages of present IDS in Grid environment, the system detects intrusions to the Grid by the coordination of the load balancing algorithm developed from the modeling of Grid and graph theory. The load balancing algorithm makes all modules of GOIDS coordinative working in Grid environment, decrease the total load of the system and increase the overall performance of GOIDS. This system is characterized by strong extension, load balancing algorithm and self-adaptability. As a result, it better solves the problem of IDS working in Grid environment, and is to a certain extent covering the shortage of the present Grid security technology.

The system works through web service, appropriate to OGSA. It has been applied to the Tianjin Suburban Industrial Grid, providing intrusion detection service for the grid. This application has proved the effectiveness of the system and its merits such as strong extension and load balancing.

Keyword: Grid, Intrusion Detection, load balancing algorithm, Webservice

图 目 录

图 2.1	GSI 结构.....	7
图 2.2	GOIDS 结构.....	12
图 2.3	GOIDS 功能流程图	16
图 2.4	信息采集模块	18
图 2.5	安全警戒模块	20
图 2.6	信息共享模块	21
图 2.7	信息分析模块	23
图 2.8	GOIDS 分布示意图	26
图 2.9	网格图结构	28
图 2.10	安全警戒模块与信息采集模块负载均衡算法.....	30
图 2.11	信息分析模块与信息共享模块负载均衡算法	31
图 2.12	GOIDA 解决方案.....	33
图 2.13	警戒-采集关联	38
图 2.14	警戒-采集关联结果.....	40
图 2.15	分析-共享关联	41

图 2.16	算法结果	42
图 3.1	GOIDS 系统构架.....	49
图 3.2	数据接口	50
图 3.3	学习数据集获取流程	52
图 3.4	网络速度探测接口工作示意图	53
图 3.5	安全警戒模块运行效果图	55
图 3.6	安全警戒模块运行效果图	55
图 3.7	信息采集模块警报信息图	56
图 4.1	系统负载比较	63

表 目 录

表 2.1 警戒—采集关联运算过程 39

表 2.2 分析—共享关联运算过程 41

表 4.1 测试节点环境列表 61

表 4.2 测试结果..... 62

南开大学学位论文版权使用授权书

本人完全了解南开大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版；学校有权保留学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：孟晋津

2008 年 5 月 28 日

经指导教师同意，本学位论文属于保密，在 _____ 年解密后适用本授权书。

指导教师签名：		学位论文作者签名：	孟晋津
解 密 时 间：		年	月 日

各密级的最长保密年限及书写格式规定如下：

内部 5 年（最长 5 年，可少于 5 年）
秘密★10 年（最长 10 年，可少于 10 年）
机密★20 年（最长 20 年，可少于 20 年）

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名： 应晋津

2008 年 5 月 28 日

第一章 导论

第一节 论文背景

1.1.1 论文背景

网格 (Grid)^[1] 技术是伴随网络技术快速发展而提出的一种新兴技术, 其最初提出的原型为计算网格, 初衷是将分布在各地的超级计算机通过网络联合以进行大规模的科学计算, 并且可以通过调度算法对任务进行规划与调度, 以达到充分利用网络中的闲置运算能力的目的, 这种强大的处理能力和资源动态分配的设计思想是传统分布式系统所不能比拟的。因此, 网格技术从一提出就受到了世界各国研究人员和机构的关注与重视, 并掀起了信息技术的第三次浪潮。随着网格技术的不断发展, 网格已经从单一的计算网格发展成为包括计算网格、数据网格、知识网格、服务网格多种类型的网络服务技术, 向用户提供强大的“无缝集成和协同”服务。但与此同时, 网格的安全问题也不容忽视, 传输信息的窃取、用户身份的冒用、分布式拒绝服务攻击 (DDOS) 等等网络攻击手段都会成为影响网格正常使用的威胁, 并且, 由于网格的开放性、动态性、多样性的特点, 导致同样的安全问题在网格中处理与防范都要比传统网络和分布式系统更复杂。由此可见, 网格的安全是一个极具挑战的问题。

目前, 网络安全问题的主要解决方案是通过部署 Globus 项目中的 GSI(Grid Security Infrastructure)软件包来实现。GSI 提供了基本的安全认证和安全通信等服务。但是网格作为一个动态且多样的系统, 仅仅通过这些保密性、完整性的保障是不够的。网格结构的变化、接入网络的计算机系统及应用软件的多样性所导致的各种系统漏洞、来自外部网络的拒绝服务攻击, GSI 都无法提供有效的方法对网格进行保护, 因此, 一个对网格进行有效监测、结构良好、可扩展性强、动态配置性好的入侵检测系统 (Intrusion Detection System, IDS) 将对网格起着极其重要的作用。对网络安全以致网格技术的发展都有重大意义。

《天津市郊区企业信息网》是科委、信息办支持的项目, 旨在通过建设服务网格环境为天津市 3000 余家郊区企业提供信息共享、行政服务、供求链服务、

产品研发辅助等多方面服务。论文作者参与该项目的系统构架与功能设计工作和部分功能的实现。

1.1.2 论文主要工作

本文从网格特点入手，提出了一种适合于网格环境的入侵检测模型及相应的负载均衡算法，根据天津市郊区企业信息网的网格环境进行系统实现与配置，并对系统的负载均衡算法进行了实验，本文的主要工作有：

- 1) 分析入侵检测技术与现有网络安全技术进行了研究这两类技术分别在解决网络安全问题中的不足。
- 2) 通过分析了网格中常见的安全问题与入侵方式，提出网格中入侵检测系统的设计需要。
- 3) 提出面向网格的入侵检测系统 GOIDS，对该系统整体构架及系统中各模块进行了详细的分析与设计。
- 4) 针对 GOIDS 并结合网格环境特点提出负载均衡算法 GOIDA，并给出一个 GOIDA 的解决方案。
- 5) 对 GOIDS 进行了实现，介绍了实现中应用到的技术，并介绍了 GOIDS 的系统构架与接口实现方式。
- 6) 根据天津市郊区工业信息网的网格环境特点对 GOIDS 进行了相应的设定，并对系统的网格环境适应性，扩展性与负载均衡能力进行了实验。

第二节 本文内容组织

本文根据目前的研究和发展方向，通过对网格环境中入侵行为的研究，针对国内外对网格环境入侵检测方面研究的不足，提出了一种面向网格入侵检测模型，该系统通过 Web 服务和 XML 技术与开源入侵检测软件 Snort 相结合，重点对网格中入侵检测系统可扩展性与负载均衡问题进行了研究，该系统在项目《天津市郊区工业信息网》中进行配置，最终对系统的可行性与入侵检测能力进行了分析。本文共分五章，组织安排为：

第一章为本文导论，介绍本文所涉及的课题研究背景与工作整个论文的章节安排。

第二章通过分析网格中入侵行为的特点及网格环境安全的需要,针对入侵检测系统与现有网络安全技术的发展现状以及在解决网络安全问题时的不足,提出了一种面向网格的入侵检测系统(GOIDS),并对系统的体系结构、各模块的功能与设计进行了详细的介绍,随后针对GOIDS与网格环境的特点提出一个负载均衡算法,并给出该算法的一个解决方案。

第三章结合GOIDS的实现工作对实现过程中使用到的应用技术进行了介绍,给出了GOIDS的系统构架,对系统中主要接口的实现方法进行了讲解。

第四章介绍了天津市郊区工业信息网的网格环境特点,并针对该网格环境对GOIDS进行了配置,并通过实际使用中的数据采集与运行状况记录验证了该系统适应网格环境,在网格中具有良好的有效监测能力、扩展能力以及负载均衡能力。

第五章是全文的总结以及对以后工作的展望。

第二章 面向网格入侵检测模型的研究

在传统网络环境下的入侵检测系统和网格技术都有一些成熟的理论和实际运行案例，但面向网格的入侵检测系统的研究还处在起步阶段，至今还未出现被广泛接受的系统模型，已提出的几个系统都存在缺少扩展能力、自适应能力差、没有考虑负载均衡等问题。本章通过分析网格环境中特有的入侵行为，研究入侵检测系统与网格安全技术发展现状，针对入侵检测系统与网格安全技术为解决网络安全问题中的不足，结合网格结构特点以及网络安全的需求，提出了一种面向网格的入侵检测系统模型（Grid Oriented Intrusion Detection System, GOIDS），并针对该系统提出一个负载均衡算法。通过负载均衡算法的调度，GOIDS 可以对网格环境进行入侵检测保护，并具有扩展性强、动态负载均衡、拥有自适应能力的特点。

第一节 问题分析

2.1.1 网格环境安全问题分析

网格计算注重于分布在不同地理位置的各种资源之间的共享、协作与有效使用。相比传统的分布式计算，具有更好的开放性、动态性、多样性的特点，网格将分布在网络中各种计算资源、存储数据、硬件设备、服务机制、应用软件联系整合到一起，并对资源与用户的变化进行动态管理，这就使网格系统的安全问题更加复杂，比如体现在以下几个方面：

- 1) 网格环境中用户与资源数量可能会很大，而且在不断变化，各个组织与个人会根据自身需要进入网格提供资源或者索取服务；
- 2) 网格环境提供的服务进程需要动态的申请、执行与结束，这就使相应的资源也需要动态的申请、使用与释放，而且同一服务进程在执行期间也可能更换不同的资源；
- 3) 资源可能来自不同的组织和系统，各种资源使用不同的认证、授权机制与安全策略，这些机制与策略难以改变，这使申请这些资源更为复杂；

- 4) 这种资源可能分属不同的自治机构，共享这些资源可能会触及到版权、费用等方面问题；
- 5) 由于使用费用的原因，用户使用资源时可能需要在不同的资源上映射帐号与身份证明；

与此同时，由于网格环境具有强大的计算资源和巨大的存储能力，这对入侵者是一个很大的诱惑。网格环境开放性、动态性、多样性的特点又为入侵者提供极为便利的条件，有利于入侵。更有甚者，可能利用网格中强大、海量的计算资源对其他系统或者就是网格本身进行攻击。通常网格环境中的入侵方式有以下几种情况：

- 1) 非授权访问：入侵者通过猜测、盗取及通过合法用户日常中的疏忽得到密码或通过盗取合法用户的私钥文件等方式，伪装成合法的网格用户对网格发起攻击。
- 2) 误用与滥用：网格合法用户或者非授权用户滥用权限时，就会出现违反网络安全策略及越权行为。这种入侵可能是一种具有侵略性的网格资源恶意使用，也可能是用户的误操作，所以这类入侵行为需要根据网格的安全策略来进行判断。
- 3) 网格漏洞攻击：由于网格的动态性与多样性特点，网格环境中节点可能是各大院校、科研机构的大型计算机，也可能是分布于网格各个位置的各类工作站与服务器，还有可能是个人电脑及移动设备。各种设备提供的安全保护级别不同，系统可能存在的漏洞也多种多样。这使入侵者可以通过软件工具或程序寻找到系统、网络协议、应用程序中的弱点从而对网格系统进行攻击。出现的形式多为拒绝服务攻击（DOS）、蠕虫攻击等。
- 4) 基于主机与基于网络的攻击。以上三点中的攻击行为都是针对网格环境特有的攻击。在网络中普遍存在的基于主机或者基于网络的攻击行为对网格环境同样具有很大的威胁。

现今网格系统中安全保障机制主要是通过 GSI 提供安全认证与安全通信来实现。但由于网格是一种开放式环境，仅仅采用 GSI 提供的安全机制是不够用的，比如入侵者可以通过窃取保护用户私钥的加密文件对系统进行非授权访问，或通过 DDoS 攻击网格中的服务器导致系统瘫痪，GSI 对这些攻击都无法进行有效的防护，因此在网格环境中配置具有更强大防御能力的入侵检测系统（IDS）是必不可少的。本文针对现有网络安全机制的不足，研究面向网格入侵检测系

统，对网络安全及网格技术的发展具有重要的意义。

2.1.2 入侵检测与网络安全技术发展现状及不足

1. 入侵检测技术发展现状及不足

入侵检测系统作为保障网络安全的重要手段，一直受到广大中外学者与研究机构的关注。由于一些固有的缺陷无法进行改造和弥补，传统的单机入侵检测系统已经无法适应当今规模日益扩大的网络环境以及更多样的攻击手段，例如在拒绝服务攻击（Denial of Service, DoS）基础上发展而来的分布式拒绝服务攻击（Distributed Denial of Service, DDoS），就是通过分布式的手段向网络中发布海量的数据包，消耗系统资源，导致系统瘫痪，使合法用户无法获得资源，这种攻击是传统单机入侵检测系统无法防范的。于是，出现了分布式入侵检测系统，并成为现在国内外入侵检测系统研究的热点。许多成功的分布式入侵检测系统被开发出来，其中包括^[5]：

(1) 20 世纪 90 年代美国加州大学 Davis 分校提出 DIDS(Distributed Intrusion Detection System)系统，该系统采用分布采集、集中处理的方式，通过将采集到的主机与网络数据汇总到中心节点，进行统一处理，但中心节点的处理能力成为了整个系统的瓶颈所在，而且当中心节点受到攻击时可能会出现整个系统瘫痪的情况。

(2) 1996 年美国 Texas A&M 大学提出了 CSM(Cooperating Security Managers)系统，该系统采用了对等体的组织方式，每个 CSM 就是一个完整的入侵检测系统，多 CSM 协同工作就形成了一个分布式入侵检测系统，这样的设计弥补了 DIDS 中心节点为瓶颈的弱点。但该系统在 CSM 数量增大的情况下，出现了交互信息量剧增以及检测能力不强的问题。

(3) 2001 年日本 IPA(Information technology Promotion Agency) 提出 IDA(Intrusion Detection Agent system)系统，该系统采用两层的系统框架，并由多主机协同检测，并采用移动代理技术对信息进行自动收集。但是，该系统只定义了某类特定入侵事件，因此只适用于检测某一类分布式入侵，扩展性不强。

(4) 在国内，对于分布式入侵检测系统的研究也有了一定的成果，如大规模分布式入侵检测系统(Large-scale Distributed Intrusion Detection System, LDIDS)采用分布采集、动态协调、集中管理的思想。采用树型的分层体系结构设计了

一种大规模的分布式入侵检测系统。

这些系统已经受到广泛认可，并且已经有实际系统投入使用。但传统入侵检测系统没有通过负载均衡的思想对系统进行设计，使系统存在处理数据集中、应变能力差、处理速度较慢、扩展能力差等问题，无法适应网格环境动态多变的特点，在网格环境中无法提供有效的入侵检测与处理。现今网格中入侵检测系统研究都是在传统网络入侵检测系统的研究基础上进行的，通过使用传统入侵检测系统的检测技术、手段以至使用传统入侵检测系统的工具和软件，结合网格环境特点研究与提出适合网格的系统构架，是现今网格中入侵检测系统研究的主要方向。

2. 网络安全技术发展现状与不足

1) GSI

当前网格技术的发展还处在起步阶段，正面临着由技术研究向商业服务的转变，这使网络安全的研究日趋重要。目前主要是通过部署 Globus 项目中的 GSI (Grid Security Infrastructure, 安全基础结构) 为网格提供安全保证，GSI 拥有 CA 认证和 X.509 认证，为网格提供了公钥系统、数字认证系统以及授权许可和单一标志等安全机制。GSI 的结构如图 2.1 所示，其中 PKI (Public Key Infrastructure, 公钥基础结构) 与 SSL (Secure Socket Layer, 安全套接层) 是网格主要安全技术的基础^[6]。

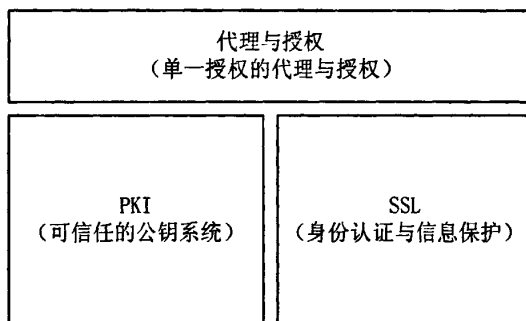


图 2.1 GSI 结构

- PKI: 是当前应用最广泛的网络安全认证技术，它建立在公钥密码学基础上，提供了加密、数字签名等密码服务及所必需的密钥和证书管理体系。完整的 PKI 系统是由 CA (Certification Authority, 证书权威)、数字证书库、密钥备份及恢复系统、证书作废系统、应用接口 (API) 等基本构成部分。

其中 CA 是 PKI 的核心，它负责数字证书的申请及签发，必须是具备权威的机构。

- SSL：是一种建立在 TCP/IP 套接字（Socket）基础上，采用 RSA 算法的安全通信技术。IETF 机构又对 SSL 进行了标准化工作，提出 TLS（Transport Layer Security）。SSL 通过接收方发给发送方一段随机消息，发送方使用自己的私钥将信息加密，然后连同本方的公钥将加密的信息发送给接收方，接收方通过公钥解密信息，如与原先随机消息一致则通过认证。对调双方角色同样通过认证后双方就可进行加密的通信。

2) 网络安全技术发展趋势

当今网络安全保证只是通过 GSI 提供的基本安全认证与安全通信来实现的。GSI 提供了在网格计算环境中的安全认证；支持网格环境中主体的安全通信；支持跨虚拟组织的安全；为网格通信提供保密性、完整性；为网格用户提供单点登陆和权限委托的能力。但网格作为一个多平台、开放、动态的系统仅依靠 GSI 是远远不够。针对网格结构的变化、各种系统的漏洞、拒绝服务攻击，GSI 都无法提供有效的方法对网格进行保护。全球的网格研究者正在为网络安全技术发展进行着不懈的努力，其发展方向可分为下面三类^[7]：

- 入侵检测方向：网络安全的目标是要求网格在遭受到攻击的情况仍能持续不断的为用户提供服务，然而网格作为一个开放型系统仅提供安全访问与信息保护机制以防止用户非法访问及信息窃取是远远不够的。研究网格环境下的入侵检测技术，使之成为可以在网格中实际应用的技术，减少非法入侵对网格系统的伤害是网络安全研究的一个大方向。
- 动态控制方向：网格环境具有动态性强的特点，不同的网格节点可能使用不同的操作系统，对安全性的要求也不相同。同时网格中运行的作业对安全性要求也存在差异，如一些高度敏感的应用可能需要更为复杂的安全机制保护，但完全使用复杂的安全机制会对网格的运行产生不良的影响，对于安全要求不高的应用这将对网格资源的一种浪费，增加了网格的负担，所以需要针对网的实际需要动态控制网的安全机制。
- 持续监控方向：网格需要对存在于其中的虚拟组织进行持续监控才能保证网的安全。这是由于虚拟组织中的单一成员改变其安全策略时会导致整个虚拟组织的安全策略发生变化，而对于整个网来说只有整个虚拟组织安全策略发生变化并产生效果后才能检测出来，这将使网环境无法适应

这种改变，导致网格系统无法提供良好的安全保证。所以对网格中虚拟组织提供持续性监控是很有必要的。

3) 网格中入侵检测系统发展现状及不足

当前网格技术的发展还处在高速发展阶段，正面临着由技术研究向商业服务的转变，这使网格安全的研究日趋重要。目前主要是通过部署 Globus 项目中的 GSI (Grid Security Infrastructure, 安全基础结构) 为网格提供安全保证。但网格作为一个开放环境，具有结构多样、动态性强的特点，只通过安全认证与安全通信机制进行安全保护是不够的。研究在网格中的入侵检测系统是网络安全技术发展的一个重要趋势，已经有一些学者进行了初步的研究，提出一些模型，但至今没有系统模型被广泛认可，网格中入侵检测系统的研究还处在不断探索的阶段中，已经提出的系统主要有：

(1) Choon O, Snajjudin 提出了一种基于网格的入侵检测系统构架^[8]，该系统中的代理 (Agent) 分布于各个网格节点收集的审计数据并发送给分析服务器，这种集中式的解决办法消耗了大量的处理时间和存储空间，并且扩展性较差；

(2) Tolba 等提出 GIDA (The Grid Intrusion Detection Architecture)^[9] 构架的入侵检测系统原型，该系统使用多个入侵检测分析服务器之间协同工作的方式来发现入侵行为，解决了伸缩性问题，但未考虑负载均衡情况；

(3) Kenny S, Coghlan 等提出的 R-GMA (Relational Grid Monitoring Architecture)^[10] 采用模块化设计的方式，以 Snort 作为检测模块核心，具有很好的扩展性，并能有效地存储并获取网格节点的数据，但该系统没有提出如何使用数据识别新的入侵行为，并且该系统未对考虑负载均衡问题；

(4) Fangyie L, JiaChun L 等提出的 PGIDS (Performance-based Grid Intrusion Detection System)^[11] 采用了功能模块的设计方法，通过预测数据采集点的数据流文件大小对系统进行负载能力的均衡，但没有对数据检测点的执行能力、网络速度情况和网格中节点间的关系对系统负载的影响进行考虑；2008 年 4 月该文作者又提出的 DGIDE (dynamic grid-based intrusion detection environment)^[12]，该系统对扩展性和负载能力进行了进一步的研究，但同样是只考虑了采集点的数据流文件大小对负载能力的影响，缺少对网格中其他的因素的考虑。

总体看来，当前网格中入侵检测研究还具有许多的不足，主要表现在无法适应网格动态性强的特点。具体表现在结构、自适应能力与负载能力方面上：

- 结构上缺乏良好的扩展能力，网格发生变化时难以对变化的网格节点进行

有效检测；

- 网格环境的多变使攻击方式、攻击切入点等经常发生变化，而现今大部分网格入侵检测系统研究都没提供自适应能力，使系统的检测能力无法适应网格环境变化，无法对新的攻击行为进行响应；
- 虽然现今研究网格中的入侵检测系统很多采用了层次模型和区域划分方式实现，但都没有考虑到组件的负载均衡问题，这将使系统在网格环境运行时极易产生瓶颈，影响系统的入侵检测能力。

因此，本文认为负载均衡问题是网格中入侵检测系统研究的一个关键核心问题，也是网络安全问题研究的一个重点，是有效提高网格环境抵御入侵的一个研究角度。正是针对现有网格中入侵检测系统研究的缺点，本文开展了面向网格入侵检测技术的研究，尤其对它们的核心问题——负载均衡问题方面作了算法和实现的一些工作，通过基于负载均衡的面向网格入侵检测的研究解决网格环境的安全性问题，这对网格的实际使用很有必要，且对今后网格技术发展极具意义的。

2.1.3 网格环境入侵检测系统需求

通过对前面章节网格环境中出现的安全问题的研究，结合网格环境开放性、动态性、多样性的特点，并针对现有网格安全技术及现有网格中入侵检测系统研究的不足，本文提出一种面向网格的入侵检测系统架构(Grid Oriented Intrusion Detection System, GOIDS)，该系统具有下面 4 个特点：

- 1) 有效检测：GOIDS 的首要功能就是要实现对网格环境中的入侵检测行为进行有效检测，针对网格环境中出现的安全问题，GOIDS 既要实现对一般系统的入侵行为的监测，也要对网格环境特有的安全入侵行为提供监测能力。就是要检测出网格中的非授权访问、误用与滥用、网格漏洞攻击以及基于主机与基于网络的攻击；
- 2) 结构良好：模型以功能模块为基本单元，通过层次结构实现，层次间通过接口实现互联。这使该模型可根据网格的发展变化或根据特定网格的实际需要对模块进行修改和替换，使 GOIDS 具有良好的可维护性和升级能力。同时也可对一种功能模块提供多种解决方案，并根据用户需求或实际网格环境状况进行多种方案的配置组合，以提供更多的安全解决方案；

- 3) 可扩展性强：由于网格用户与网格资源可能在不断增加，所以需要模型提供极好的可扩展性，以适应网格规模的不断扩大；
- 4) 具有负载均衡能力：针对网格环境的动态性，网格中作业具有资源动态多变的特点，如何动态配置 GOIDS 的资源以达到监测资源的最优化配置成为该模型的一个核心机制，模型通过对网格环境模型化和基于图论理论的负载均衡算法对 GOIDS 进行有效配置。

第二节 面向网格的入侵检测模型体系结构

针对已有网格中入侵检测系统研究的成果以及已提出的系统不足，本文提出 GOIDS 模型，模型采用层次结构、模块化设计方式。本节将对该模型的结构、实现功能、各模块的功能与设计进行详细介绍。

2.2.1 模型体系结构

本文提出了一种采用模块化设计方式的面向网格的入侵检测模型 (GOIDS)，该模型以功能模块作为最基本单位，网格中心节点通过 GOIDS 的负载均衡算法对 GOIDS 各模块的运行进行了调度，使 GOIDS 的模块以最优化的组合形式协同工作，以达到检测最优化的目的。从已有的研究成果中来看，GIDA^[9]、R-GMA^[10]、PGIDS^[11]、DGIDE^[12]采用了模块化设计方式，分化了入侵检测系统中的各个功能，各模块根据功能不同分别形成多个层次，采用这种设计方式可以使系统具有较好的扩展能力，并为系统各模块负载均衡的实现提供了方便。但 GIDA^[9]、R-GMA^[10]采用检测模块与采集模块相对固定的组合方式，这使系统的负载均衡能力较差；PGIDS^[11]、DGIDE^[12]考虑到负载均衡问题，将系统中采集与检测功能进一步分化，但没有对采集到的数据进行再学习，使系统无法对新类型的攻击进行响应，并且只对检测文件大小对负载均衡效果的影响进行了研究，未对网格中其他因素进行考虑。

针对已有的研究成果与不足，GOIDS 将包含五种功能模块，分别为：信息采集模块、信息共享模块、信息分析模块，安全警戒模块、报警通道，五种模块协同工作实现 GOIDS 入侵检测与信息学习两大功能，使系统在提供较好扩展与负载均衡能力的同时，提供了对采集到的数据进行再学习以响应新类型攻击

的功能。GOIDS 的所有模块都在网络中心控制节点进行注册，在中心控制节点上产生一个所有模块的映射表，由网络中心控制节点实现 GOIDS 负载均衡算法，以及对模块的运行状态与模块间信息共享进行维护。五种模块根据功能的不同分别构成 GOIDS 的三层：信息收集层、信息传输层、信息处理层，结构如图 2.2 所示。

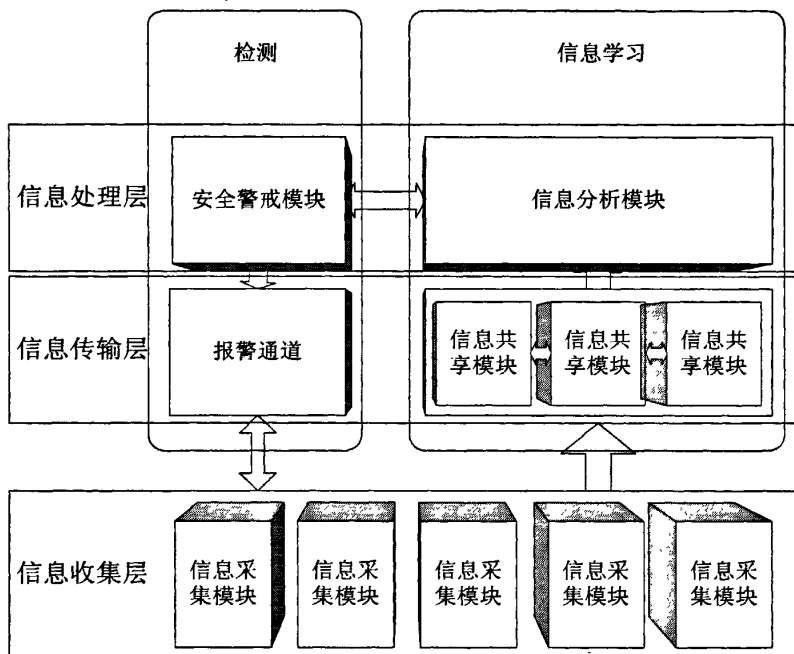


图 2.2 GOIDS 结构

可在布置网络节点时根据节点的运算与存储能力选择 GOIDS 的不同功能模块，这样可以在已形成的网络中形成入侵检测系统而无需增加额外的设备，就可以为网络运营提供安全服务的保障。功能模块作为网络入侵检测服务提供者的同时又是运行在节点上的应用程序使用着网络中的资源，网络为运行在其中的应用程序提供强大的运算与存储资源，这使 GOIDS 的入侵检测能力有了更好的保证。

根据功能模块组成的形式不同，GOIDS 可分为两种运行模式：

1) 基本检测模式：网络的每个节点都要包含一个信息采集模块，一部分节点包含安全警戒模块，网络中心控制节点通过 GOIDS 的负载均衡算法对这两种模块产生关联为网络环境提供最基本的面向网络的入侵检测功能（GOIDS 负载

均衡算法与模块间关联关系将在本章第三节作详细解释)。该种模式下 GOIDS 不提供自学习功能,系统只会以默认设置的安全规则对网格环境进行入侵检测。该模式必需的模块为:信息采集模块、安全警戒模块、报警通道。

2) 完全模式:该模式包括基本检测模式,即包含有与基本检测模式相同的信息采集模块、安全警戒模块与报警通道。同时网格中还要至少包含有一个信息分析模块与至少一个信息共享模块,以实现 GOIDS 的自我学习功能。GOIDS 可以通过机器学习的方式对网格中检测到的审计数据进行自学习,这样就使系统对未知类型的攻击有了检测能力,最终通过对安全规则进行更新使整个系统对新类型攻击都可进行检测,这样结合了异常检测与误用检测的优点,让 GOIDS 不但在结构上做到面向网格,更在检测能力上适应了网格开放、动态、多样的特点。完全模式下五类模块均要包含,同时实现了 GOIDS 的检测与信息学习功能。

综上所述可以看出,基本检测模式是完全模式的一个组成部分,完全模式在基本检测模式的基础上增加了对审计数据进行机器学习并发现攻击的功能,即实现传统入侵检测系统提供的异常检测功能。GOIDS 采用两种运行方式,这是由于网格环境复杂多样,自学习功能需要较强的信息存储与处理能力,有些网格环境中可能无法提供信息自学习功能所需要的运算与存储资源,这样 GOIDS 就只能运行在无学习功能的基本检测模式下,只提供基于对网格环境的入侵检测功能,无法提供自学习功能对新类型的攻击进行响应。

图 2.2 上 GOIDS 中的五种功能模块所实现的功能不同,其中 a)类功能表示运行在两种模式下均必需的功能,b)类功能表示只需要在完全模式下提供的功能,即基本检测模式需要 a)类功能,完全模式需要 a)+b)类功能。

● 信息采集模块,被设置在网格环境中的每个节点上,结合基于主机和基于网络的入侵检测技术,负责对该节点的主机信息、网络信息进行收集与封装,发送给信息共享模块及通过报警通道发送给安全警戒模块,并最终对报警通道送来的安全警报进行响应。

实现功能:

- a) 将所在网格节点的信息通过报警通道即时发送给安全警戒模块,进行入侵检测;接收报警通道送来的安全警报;
- b) 将所在网格节点的信息封装,定期发送给信息共享模块,作为系统自学习的样本;

● 信息共享模块，负责对信息采集模块发送来的数据包进行处理，通过负载均衡算法为信息分析模块提供学习数据集。

实现功能：

b) 接收信息采集模块发送的数据，对数据进行处理，形成学习数据样本库；通过负载均衡算法为信息分析模块提供学习数据集，以进行系统的自学习。

● 信息分析模块，负责对信息共享模块中的数据进行统计、分析与学习，根据网格使用中数据的特点对网格环境中入侵检测的规则进行更新，并将规则通知给安全警戒模块，信息分析模块可以采用多种方式实现，包括数据挖掘技术、人工神经网络技术、支持向量机技术等，用户可以根据实际需求对模块进行选择。

实现功能：

b) 通过机器学习技术对信息共享模块提供的样本数据进行自学习；当产生新的安全规则时通知相应的安全警戒模块，并对其规则进行更新；

● 安全警戒模块，根据入侵检测规则负责对各个信息采集模块的数据进行检测，当发现入侵时通过报警通道及时通知出现入侵的信息采集模块所在的节点。

实现功能：

a) 对送交该安全警戒模块的数据进行入侵检测；

b) 接收信息分析模块的安全规则更新，使安全警戒模块可以对新类型的攻击进行响应；

● 报警通道，负责将信息采集模块的数据与该模块地址进行映射，并提交给安全警戒模块，当接受到警报通知时通过映射及时通知出现入侵行为的信息采集模块所在的节点。

实现功能：

a) 通过网格中心控制节点的设置，形成信息采集模块的数据与安全警戒模块的关联，将信息采集模块提交的信息即时提交给对应的安全警戒模块，以进行检测；接受安全警戒模块的报警信号，并通过与信息采集模块形成的关联及时地将报警信号发送给对应的信息采集模块；

由于完全模式提供基本检测模式的所有功能，所以后面的讨论中提到 GOIDS 都是指完全模式下运行的 GOIDS，基本检测模式作为完全模式的一个特例存在。

2.2.2 GOIDS 功能描述

GOIDS 分为两个主要功能，检测与信息学习。检测功能为网格提供负责对网格中收集到的数据进行入侵检测，发现入侵行为时通知相应节点进行响应；信息学习功能通过对网格中的采集到的数据进行处理，形成学习样本数据，使用机器学习系统对样本进行学习，对 GOIDS 中安全检测规则进行更新，使 GOIDS 能够更好的适应网格环境。GOIDS 功能如下如所示：

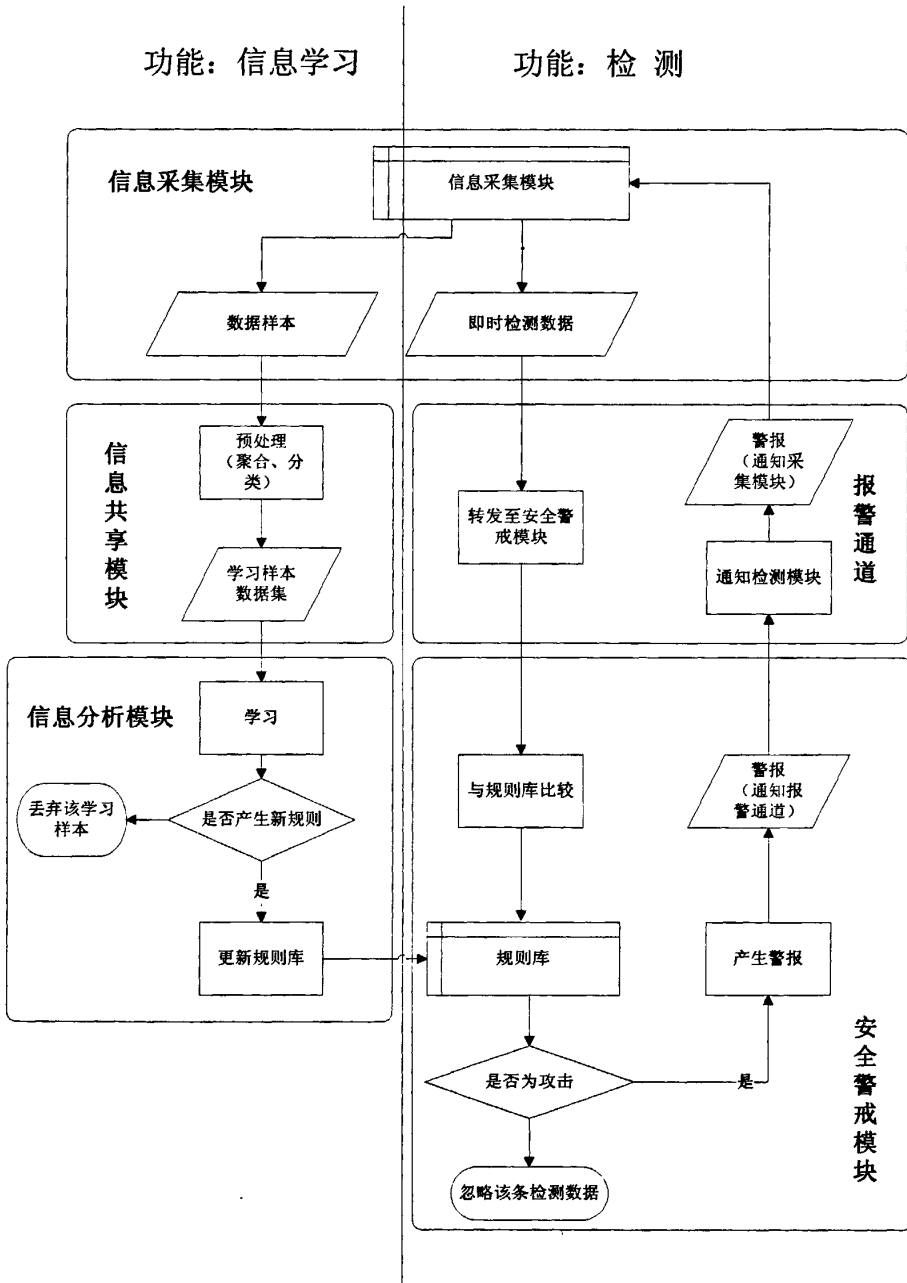


图 2.3 GOIDS 功能流程图

结合上图，以数据流程为主线，对 GOIDS 的两个功能进行描述。

- 信息学习功能：

- 1) 信息采集模块收集所在节点的信息，包括基于网络的信息与本机的信息，封装为 XML 文件发送给信息共享模块，跳转至步骤 2)
 - 2) 信息共享模块接受信息采集模块发送的 XML 文件，并对信息进行处理，形成学习样本数据集，为信息分析模块提供访问接口，跳转至步骤 3)
 - 3) 信息分析模块访问信息共享模块得到学习样本数据集，通过这些数据集为机器学习系统提供训练样本，跳转至步骤 4)
 - 4) 将学习过的数据集进行丢弃，并当机器学习系统产生新的安全规则时则跳转至步骤 5)，若无规则更新则跳转至步骤 6)；
 - 5) 对安全警戒模块中的规则库进行及时更新，跳转步骤 6)；
 - 6) 本次信息学习过程结束。
- 检测功能：
 - 1) 信息采集模块将即时的检测数据发送给报警通道，跳转至步骤 2)；
 - 2) 报警通道根据网格中心控制节点设置的关联关系将即时的待检测信息发送给安全警戒模块，跳转至步骤 3)；
 - 3) 安全警戒模块根据安全规则库对报警通道提交的数据进行检测，并丢弃该数据，当断定此攻击为入侵检测时跳转至步骤 4)，未发现入侵检测跳转步骤 7)；
 - 4) 产生警报并发往网格环境下的报警通道中，跳转步骤 5)；
 - 5) 报警通道将警报转发至网格环境下对应的信息采集模块，跳转步骤 6)；
 - 6) 信息采集模块接收到警报，做出入侵预警响应，跳转步骤 7)；
 - 7) 本次检测过程结束。

2.2.3 模块设计

1. 信息采集模块

信息采集模块包括：检测数据接口、学习数据接口、警报接口、采集引擎。

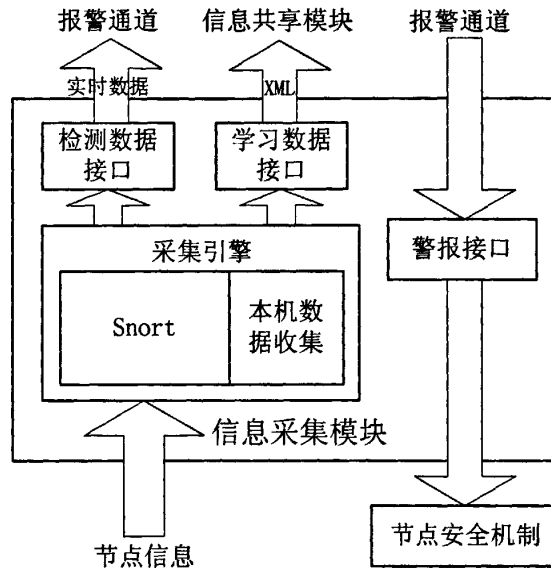


图 2.4 信息采集模块

信息采集模块的主要功能有：检测数据收集与提交、警报响应、学习数据收集三个功能。其中：

1) 检测数据收集与提交

检测数据收集的功能主要由采集引擎来完成，它负责采集基于主机与基于网络的信息，通过检测数据接口将所采集的实时数据发送给报警通道。

为了满足 GOIDS 模型有效监测的需求，信息采集模块同时收集基于主机与基于网络的信息。主机信息进行本地分析，提交给安全警戒模块的只是本地分析后确认是异常的信息；基于网络的信息是指信息采集模块采集到的所在节点的网络报文信息。这样将不必要的本机正常数据信息屏蔽掉，减少了数据传输量，同时又将基于主机与基于网络的检测方法结合起来，使本机出现的异常检测信息与当时网格环境中的网络行为相关联，以提高入侵检测系统对攻击和误用的判断能力，使安全措施的实施更加有效。

主机信息：主机节点的异常信息，包括进程异常与日志异常。进程异常：某进程 CPU 使用率、内存占有率过高的情况；日志异常通过系统自带的日志分析程序得到日志的异常状况，如文件越权访问、用户非法登陆等；

网络信息：通过分析网络数据包，从中得到连接的源地址、源端口号、目标

地址、目标端口号、协议类型、协议属性、连接状态、传输数据量等。为减少系统的复杂度，加强系统的可用性，GOIDS 采用 Snort 对网络数据进行收集，Snort 为一款成熟的跨平台开源分布式入侵检测系统软件，其最小模式下只包含一个程序文件，只提供网络数据收集的功能，正符合 GOIDS 信息采集模块跨平台、运行需求资源低的要求。

由于入侵检测即时性的要求，信息采集模块采用随采随发的方式将实时收集到的上述两类数据信息进行封装发送给报警通道。

2) 警报响应

每个信息采集模块包含有一个接受警报信号的接口，即警报接口，通过该接口，信息采集模块接受来自报警通道的警报信号，并通知本节点上相应的安全软件或安全机制产生反应。

3) 基于学习的数据收集

对学习数据的收集作为信息学习功能的第一步，与检测数据的提交有一定区别，这是由于学习过程对数据的即时性要求并不高，但需要数据经过一定的处理，所以这里不采用随采随发的方式，而是将收集到的学习数据进行初步处理，再定时发送传输给信息共享模块。这种实现方式可有效地降低系统的复杂度，而且通过定时封装为 XML 文件形式发送可以有效减少传输的带宽需求，同时，封装为 XML 文件时已经对数据进行了格式化，且 XML 文件中包含了数据的结构信息，这为后面信息共享模块的信息预处理提供了良好的基础，降低了信息共享模块的负担。

信息采集模块将采集引擎收集到的主机与网络数据包定时（如 5 秒）或定量（如 25 个数据包）的方式打成一个文件，文件采用 XML 文件格式，通过学习数据接口发送给信息共享模块，该文件大小与当时网络的流量相关。

2. 报警通道

报警通道负责将报警信号及时准确的发送到相应的信息采集模块的功能。因为安全警戒模块与信息警戒模块采用动态的方式进行配置，对同一个网格环境进行检测时会由于网格状态的不同采用不同的配置方式，还会出现检测期间信息采集模块离开安全警戒模块或者离开网格的情况，使用简单的数据直发方式会导致安全检测模块无法保证报警信号送到采集模块。因此，将发送信号的功能从安全警戒模块的单独出来形成单独的报警通道模块，并采用安全警戒模块与报警通道一一对应的方式，设置在同一网格节点中，这样实现可以在保证传

递警报时间不增加的情况下，使系统的结构性更强，系统维护代价降低，从而使更多的资源可以用在入侵检测上。

报警通道保存有信息采集模块与安全警戒模块之间的关联关系，这个关联关系是由网格中心控制节点维护的。报警通道将这个映射关系进行缓冲存储，当安全警戒模块送来的警报信息时，通过将警报中的数据地址信息与缓存中映射关系进行比对，及时找到相应信息采集模块的地址，将警报信息转发过去。

3. 安全警戒模块

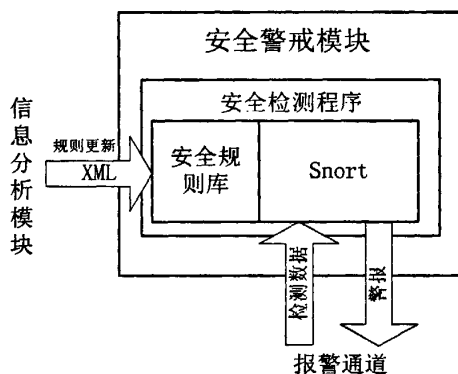


图 2.5 安全警戒模块

安全警戒模块实现两个功能：1)负责接收信息采集模块中基于主机与基于网络的检测数据，并通过安全检测程序对数据进行检测，如发现数据为攻击行为则立刻发送安全警报，通过报警通道通知相应的信息采集模块，信号信息采集模块收到警报后对本节点的安全防护机制进行通知。2)安全警戒模块中存有规则库用以比较检测数据，并接受信息分析模块发送来的新的规则完成对规则库的及时更新。

GOIDS 将采用 Snort 的入侵检测功能作为安全警戒模块的核心，直接使用 Snort 入侵检测功能作为检测核心可以在增加系统复用性、减少系统开发复杂程度的同时有效地保证了入侵检测质量，而且信息采集模块使用 Snort 数据采集组件，会使数据的传输与处理更为简单。第三章第一节“应用技术介绍”中将对 Snort 做更深入的介绍。

在传统网络环境中安全警戒模块与信息采集模块最佳配置方案是采用一一对应的关系模式，但这种配置方式并不适合网格环境，而且在网格中很难实现，

这是由于网络环境中的节点性能和操作系统多存在很大差异。进行数据采集及报警接收的信息采集模块可以实现在所有节点中，但安全警戒模块需要存储安全规则库并对检测数据进行判断，这就对部署安全警戒模块的节点的存储、逻辑处理能力与操作系统提出了要求，使安全警戒模块无法在所有节点中配置。所以运行在网格环境中的 GOIDS 最终会出现信息采集模块数量远大于安全警戒模块数量的情况。GOIDS 采用安全警戒模块与信息采集模块动态关联的方式进行配置，这种关联关系通过 GOIDS 负载均衡算法进行实现与维护，通过 GOIDS 负载均衡算法，GOIDS 将具有良好的扩展性与负载均衡能力，比安全警戒模块与信息采集模块一一对应的方式更能适应网格多样、动态的特点。GOIDS 调度算法负载均衡算法将在本章第三节“GOIDS 负载均衡算法”中做详细的说明。

4. 信息共享模块

信息共享模块的功能模块有：信息接收接口、数据库、数据共享接口、数据访问接口。

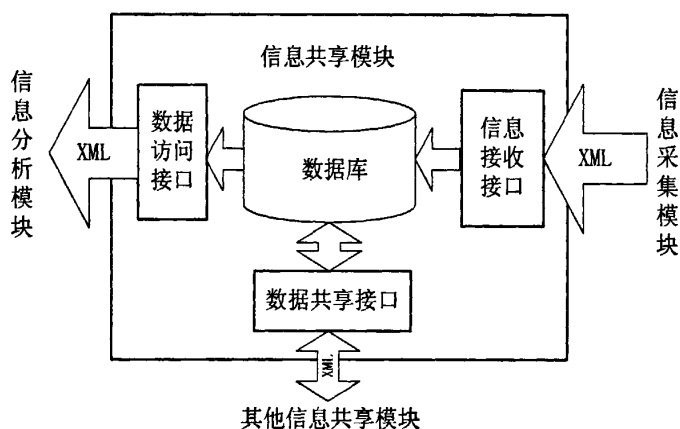


图 2.6 信息共享模块

信息共享模块通过信息接收接口接受信息采集模块传输来的 XML 文件，其中包含一定时间（如 5 秒）或一定数据量（如 25 个数据包）的节点信息，包括本机异常信息与网络数据；由于节点信息数据采用 XML 方式传输，其中包含了统一格式的信息数值、属性、结构信息，所以数据可以通过信息接收接口解析后直接存储于数据库中；信息共享模块通过数据共享接口实现多共享模块间信息的共享；通过数据访问接口为信息分析模块提供信息学习样本。

在实际使用中，系统会出现如下问题：信息采集模块存在于网格节点中，时刻都受到网格动态性的影响，会出现同一个信息采集模块在不同时间将信息发送给网格中不同的信息共享模块，或者在一段时间内某个信息共享模块接收到多个信息采集模块发送的信息。这使单个信息共享模块中的数据很难体现出网格的实际运行情况，而位于 GOIDS 中更高一层的信息分析模块希望通过统一的数据访问接口从一个“单一”的信息共享模块中获统一数据结构的学习样本，这样能在减少信息分析模块负担的情况下最大程度地正确反映整个网格的使用情况，从而提高信息分析模块的自学习能力，最终达到提高 GOIDS 入侵检测能力的目的。所以信息共享模块需要提供一个多模块之间信息共享的功能，这也是该模块名称的由来，多个信息共享模块形成信息共享集。

GOIDS 各模块之间的数据传输均采用 Web Service + XML 的方式实现，这为信息共享模块间的共享提供了极为便利的条件。网格中心控制节点中保存有 GOIDS 所有节点的映射信息，其中包括存在在网格中的所有信息共享模块的信息。每一个信息共享模块都包含有一个数据共享接口，当信息分析模块提出学习样本需求时，只需对网格环境中任意一个信息共享模块提出请求，信息共享模块根据网格中心控制节点中的信息得到所有信息共享模块在网格中的位置，然后根据负载均衡算法通过数据共享接口与其它信息共享模块进行通信，最终将所有符合要求的记录为返回给信息分析模块，形成学习样本集。这一功能的实现将在第三章第二节“系统构架与实现”中作详细讲解。

5. 信息分析模块

信息分析模块处在 GOIDS 的最高层“信息处理层”中，通过对网格中收集到数据进行学习，维护 GOIDS 安全检测规则的更新。信息分析模块以信息共享模块处理过的学习样本数据集为学习样本，其中包括了网格环境中基于本机与基于网络的数据信息，结合知识数据库中的检测规则，在自我学习系统对样本数据集进行学习训练与检测，当产生信息的安全规则时，通知对安全报警模块发送安全规则更新通知，并同时与信息分析模块中的知识数据库进行改进，以适应进一步的学习。

信息分析模块包括自我学习系统、知识数据库、用户访问接口、数据接口多个功能组件，结构如图 2.7 所示。

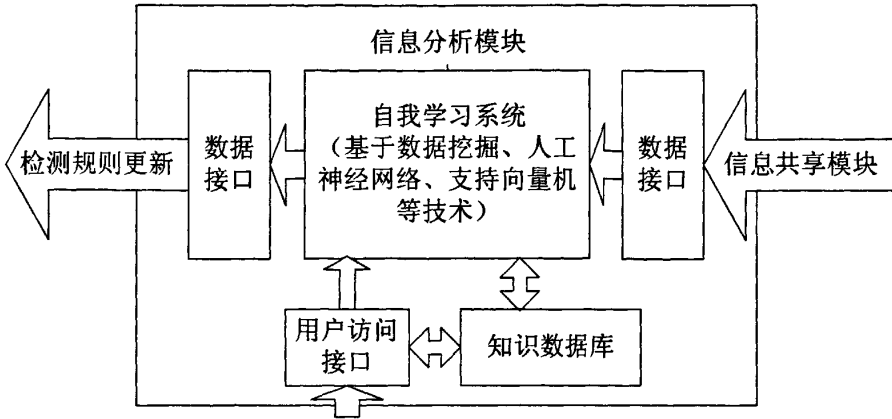


图 2.7 信息分析模块

1) 自我学习系统：自我学习系统作为信息分析模块的核心组件，提供对信息共享模块提交的检测信息进行分析、训练、学习并检测是否为攻击行为的功能。为减少系统的复杂程度，提高系统可复用性，GOIDS 中的自我学习系统采用较为成熟的计算机学习技术，可使用数据挖掘技术、人工神经网络、支持向量机技术等。GOIDS 采用功能模块的设计方法，所以可分别选用不同的自我学习系统形成各自的信息分析模块，这样有利于根据网格实际情况选择不同学习技术，以达到检测最优化的效果。

对自我学习系统的训练过程采用大量基于主机与网络的正常行为记录与成熟网络安全训练数据相结合的方式。先使用正常网络行为数据对自我学习系统，使之建立用户正常行为模型，然后通过成熟网络安全训练数据训练系统使之形成攻击模型。

2) 知识数据库：知识数据库是一个包含若干检测规则的集合。知识数据库的初始化采用信息分析模块中相应的自我学习技术生成，将成熟的样本数据送入自我学习系统中进行学习，形成一系列安全检测规则储存于知识数据库中。随着 GOIDS 的运行，学习系统将根据网格的实际检测数据生成适合于当时网格环境的入侵检测规则，新规则将存入知识数据库中。

知识数据库的规则储存包括规则数据与相关信息两部分，规则数据同于传统 IDS 中的规则，而相关信息将记录该规则生成的时间、诱因、学习系统等，这是由于网格环境具有极强的动态性，相关数据的储存将有利于规则更新的取舍，

从而优化整个 GOIDS 的性能。

3) 用户访问接口：数据访问接口为用户提供自学系统参数设置、攻击行为确定、知识数据库修改的功能。由于现今计算机自学习技术还处在研究阶段，而且大多都是以审计学为理论基础，因此检测结果误差率比较高，而且以有些理论（如支持向量机理论）为基础的模型对参数的选择与设置存在很大依赖性，所以必须人为对学习系统进行一些设置与结果判定，而且要对知识数据库拥有访问与修改能力，用户访问接口为用户有效管理分析模块提供方便，能更好的保证系统检测结果的正确性。

4) 数据接口：GOIDS 中，各模块之间采用接口方式进行通讯，使用 XML 作为传输手段，数据接口负责将模块内数据封装为 XML 文件发送出接口，或者将接受到的 XML 文件进行解析，根据模块不同生成符合模块内部使用的数据模式。接口的实现将在第三章第二节“系统构架与实现”中进行详细讲解。

第三节 GOIDS 负载均衡算法

上章对面向网格的入侵检测系统(GOIDS)的结构进行了介绍,剖析了 GOIDS 的五个组成功能模块：信息采集模块、信息共享模块、信息分析模块，安全警戒模块、报警通道。并分别对每个模块的结构、功能与设计做了详细的解释，这些模块分布在网格中的不同节点中，并由网格中心控制节点做统一调度实现协同工作。本节首先对 GOIDS 在网格中的分布结构做出分析，提出 GOIDS 负载均衡算法（GOIDA, Grid Oriented Intrusion Detection Algorithm）并进行了算法描述，最终提出一个在网格环境中 GOIDA 的解决方案。

2.3.1 GOIDS 分布结构与中心节点功能

由于网格属于松散耦合的组成方式，各个节点不存在必然的结构关系，只要能接入网络的终端设备都可做为网格中的节点存在，以图 2.8 “GOIDS 分布示意图—A.实际分布”为例，给出了网格的松散结构的示意图。图例中的网格中一共包含 16 个节点，除去一个网格中心控制节点，还有 3 台服务器，5 台 PC 电脑，两台笔记本电脑，5 台移动终端，包括 3 台 PDA 和 2 台智能手机。该网格中配置了 GOIDS，每个网格节点中都配置有信息采集模块，网格中配置了一个信息

分析模块，除此之外网络中包含 3 个安全警戒模块与 3 个信息共享模块，其中有一个节点同时包含了上述这两种模块。图中安全警戒模块用长虚线标出，信息共享模块用短虚线标出。

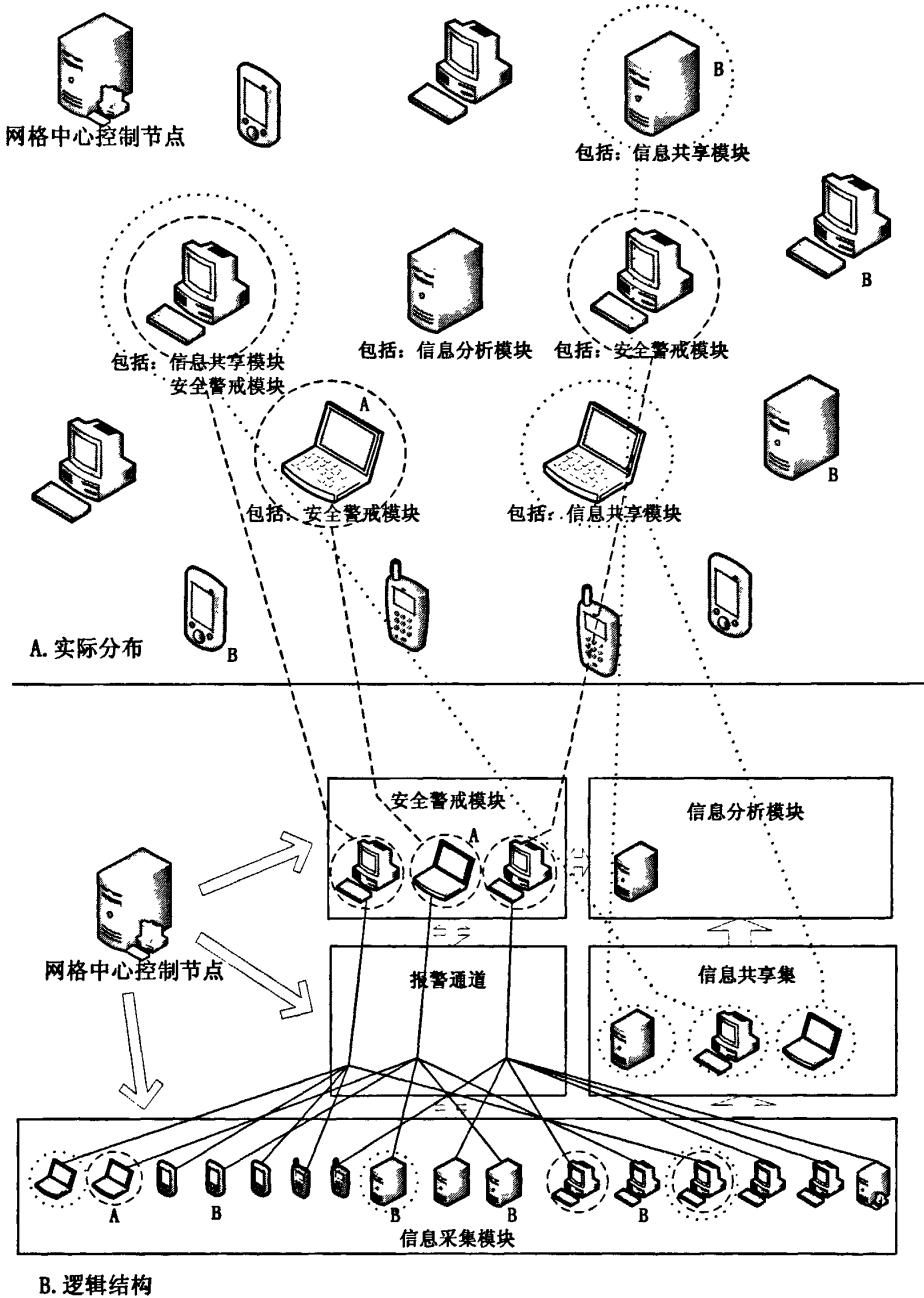


图 2.8 GOIDS 分布示意图

图中的网格包含一个网格控制中心节点，也称为网格主节点。该节点在网格中负责对网格资源与网格作业进行维护与调度，其主要功能包括：对网格资源的查看；网格资源维护，包括对网格资源的增加、减少与变更；作业提交与调度，负责用户作业的提交，并根据网格调度算法将作业分配给指定的网格节点；作业运行状态的查看。

GOIDS 作为一个运行在网格环境中的应用系统，所有模块均分布于网格节点中，其运行必然受网格中心控制节点的监控与调度。网格中心控制节点将负责：记录并维护所有存在于网格中的 GOIDS 功能模块信息，包括模块 ID、类型、所在节点位置信息，当网格中加入、减少以及变更 GOIDS 功能模块时更改相关信息；记录模块之间的关联信息，包括：安全警戒模块与信息采集模块之间关联；信息采集模块与信息共享模块之间的关联；安全警戒模块与信息分析模块之间的关联信息；根据 GOIDS 负载均衡算法（GOIDA），对模块之间的关联进行维护；GOIDS 运行状况查看。

以图 2.8-A 的网格分布为例，在图 2.8-B “逻辑结构”中表示了网格中心节点存储的 GOIDS 结构。此时网格中心节点对 GOIDS 的具体管理工作如下：

1) 网格中心控制节点维护一张 GOIDS 模块表，该表根据 GOIDS 的五类模块而分为 5 个集合，每个集合将在包含 GOIDS 模块的节点加入网格或已存在的节点中配置 GOIDS 模块都根据模块不同在模块表中进行记录，最终形成对网格中所有的 GOIDS 模块的记录。图 2.8A 网格中所有 16 个节点都配置了信息采集模块，在网格中心控制节点模块表的信息采集模块集合中对这 16 个节点都做了记录，同理，包含其他类型模块的节点也均在网格中心控制节点中做相应的记录。

2) 网格中心控制节点中还对各模块之间的关联进行管理

● 此时根据 GOIDA 的调度，图中的 A 节点，即包含安全警戒模块的笔记本电脑节点，负责所有 B 节点与本节点的安全检测工作，A 节点与所有 B 节点的实时检测信息均通过报警通道发送给 A 节点进行检测，与此同时其他包括信息采集模块的节点也与相应的安全警戒模块产生这种关联，这种关联由网格中心控制节点通过 GOIDA 进行统一分配，当发生网格环境发生变化安全警戒模块与信息采集模块之间的关联将相应产生变化；

- 网格中存在的信息共享模块在网格中心控制节点模块表信息共享模块集合中都有相应的记录。此时网格中的所有信息共享模块通过共享接口形成一个信息共享集，由于每个信息共享模块都包括一个数据访问接口，所以当信息分析模块提出学习样本请求时，向信息共享集中的哪个信息共享模块提出请求可以得到最佳的结果是一个值得研究的问题，GOIDA 对该问题提出了一个解决方案，形成信息分析模块与其最佳的信息共享模块的一个关联，并记录在网络中心控制节点中。当网格环境改变，如有新的信息共享模块加入将对这个关联重新计算与配置。

- 根据中心控制节点中记录的安全警戒模块与信息分析模块的信息，将两种模块产生关联，由于不同的安全警戒模块可能由于运行状态或者加入网格的时间不同，其规则库会出现不同，所以为区分不同的安全警戒模块，需要建立安全警戒模块与信息分析模块之间的关联关系。

3) 查看 GOIDS 的运行状态，同时信息分析模块提供用户访问接口，网格中心控制节点根据记录的节点信息对该接口产生一个映射，为用户提供更为便捷的访问方式。

从上面 GOIDS 分布结构与网格中心节点的功能可以看出，虽然 GOIDS 的功能模块分布在网络的各个节点之中，并采用松散耦合的方式运行，但通过网格中心控制节点的统一调度与分配，使各个模块之间产生关联，将所有模块组成一个有机的整体，并通过 GOIDS 负载均衡算法 (GOIDA)，将系统运行在一个更为优化的方式下，为网格环境提供更好的入侵安全检测服务。下面对 GOIDA 进行详细介绍。

2.3.2 GOIDS 负载均衡算法

1. 基本模型描述

本文给出的 GOIDS 的功能模块分布在网络的不同节点中，网络采用松散耦合的形式组成，这导致 GOIDS 的模块间不存在物理结构上的联系，只存在网络中的互连关系。

定义一个无向加权图 $G(V,E,D,W)$ 来描述这种结构，其中 V 为顶点集合， E 为边集合， D 为顶点权集合， W 为边权集合。对应 GOIDS，图的顶点 v 定义为包含有 GOIDS 功能模块的网格节点，每一个顶点都有一个唯一的 id ，顶点权 d

用来区分节点包含的不同类型的功能模块，边 e 为节点间的关系，边权 w 用量化的方式来衡量这种关系。

顶点权 $D=\{1,2,3,4\}$ ，当节点包含信息采集模块时顶点权值为 1，由于安全警戒模块与报警通道是一一对应的，所以两模块用一个顶点表示权为 2，信息共享模块权为 3、信息分析模块权 4，当一个网格节点包括多个模块时则映射为多个顶点，每个顶点对应相应模块的权，顶点间边权为 0。

边权 w 为边权集合 W 的元素，大小用 $\langle N、L、C \rangle$ 的三元组定义。

- N ：网速，根据 GOIDS 模块间的关联，网格节点与包含相关模块网格结点间的网络速度
- L ：逻辑关系，很多网格都针对特定服务或服务对象进行设计与配置，这就使网格节点之间可能存在服务上或称逻辑上的关系， L 用来衡量此种关系
- C ：负载能力，GOIDS 安全警戒模块的所在节点的负载能力

边权 $w = \alpha N + \beta L + \gamma C$ ，其中 $\alpha + \beta + \gamma = 1$ ， $0 < \alpha、\beta、\gamma < 1$ ， $\alpha、\beta、\gamma$ 分别为 $N、L、C$ 对应的权值，具体数值针对网格的实际情况进行设置，当 $N、L、C$ 某一项为 ∞ 时， w 为 ∞ ，此时说明该边连接的两顶点无法连通。

通过定义，GOIDS 所在的网格环境形成如下图的形式：

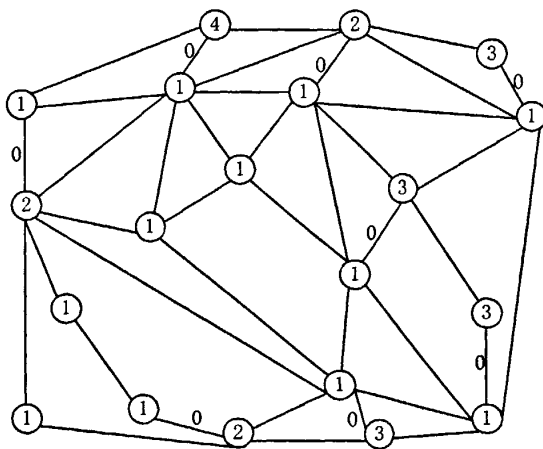


图 2.9 网格图结构

其中标记权值为“0”的边两端的顶点为同一个节点中的不同模块，所有边都包含权 w ，由于网格环境没有特定的物理结构，所以对应的图应为完全图，由

于表现形式所限图 2.9 示意图, 只表现出一部分连接状况, 实际上顶点间的连接程度远多于图上所示

2. 算法描述

定义好无向加权图的各个元素后, 再针对 GOIDS 的实际需要对图作一个简化: 由于 GOIDS 中只需要维护三种模块间的关联, 分别是安全警戒模块与信息采集模块、信息分析模块与信息共享模块、信息分析模块与安全警戒模块, 所以在图中只保留这三类关联涉及到的模块间关系以及同类模块间的关系, 其它关系均不考虑, 从图中去除。并且信息分析模块与安全警戒模块的关联只需监控网络中存在多少安全警戒模块以接收信息分析模块的规则更新, 无需 GOIDA 维护, 所以 GOIDA 只对另外两个关联进行计算与维护。

安全警戒模块与信息采集模块:

只涉及到权为 1 与 2 的顶点, 取出该类顶点, 形成一个子图, 称为检测子完全图, 安全警戒模块与信息采集模块的关联关系就转变为在检测子图中的从多个顶点引出的支撑树问题, 其中树高均为 1。算法如下:

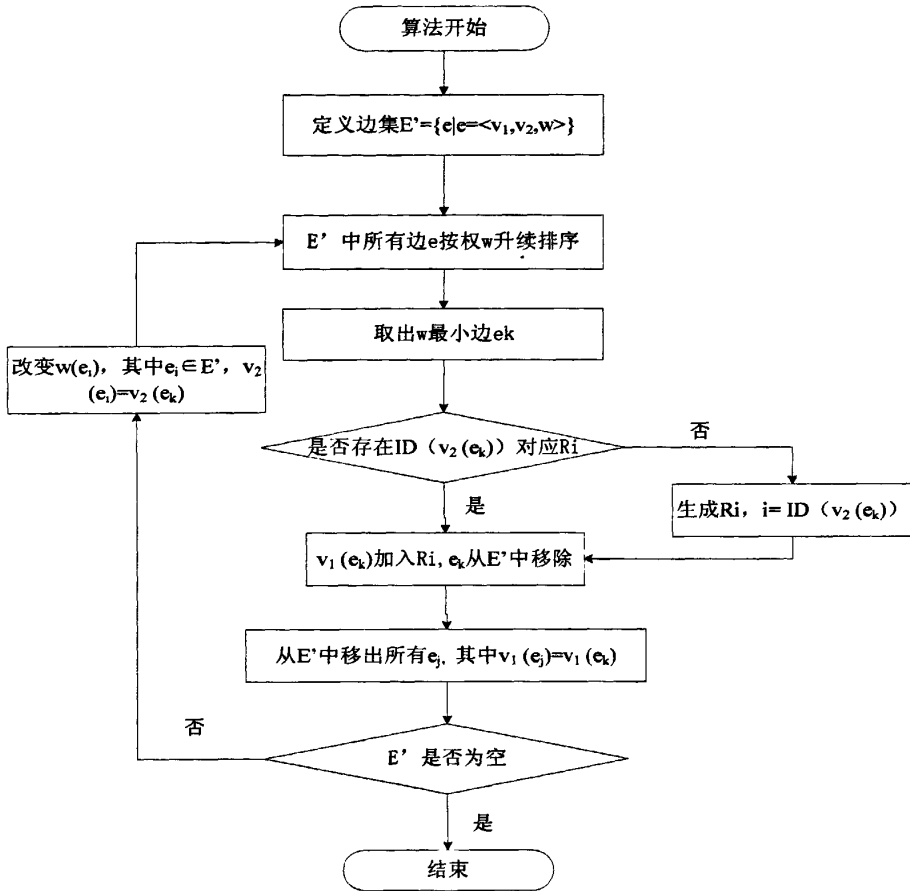


图 2.10 安全警戒模块与信息采集模块负载均衡算法

- (1) 定义边集 $E' = \{e | e = \langle v_1, v_2, w \rangle\}$, v_1, v_2 为边两端顶点, w 为边权。检测子完全图中 $d(v_1) = 1, d(v_2) = 2, d$ 为顶点权。定义结果集合 Re , Re 中包含集合 R_i , 其中 i 为 v_2 所对应顶点的标识, 用 $i = ID(v_2)$ 表示;
- (2) 将 E' 中所有边 e 按照 w 进行升序排序;
- (3) 取出权最小的边 e_k , 若存在 $ID(v_2(e_k))$ 对应的 R_i , 则将 $v_1(e_k)$ 加入到 R_i ; 若 R_i 不存在, 则创建 $R_i, i = ID(v_2(e_k))$, 并将 $v_1(e_k)$ 加入。将 e_k 从 E' 中移除, 并将 E' 中所有 $v_1(e_j) = v_1(e_k)$ 的边 e_j 从 E' 中移出;
- (4) 若 $E' = \Phi$, 算法结束, $Re = \{R_i\}$ 为结果; 若 $E' \neq \Phi$, 继续(5);
- (5) 改变 $w(e_i)$ 其中 $e_i \in E', v_2(e_i) = v_2(e_k)$ 。重复(2)(3)(4);

说明:

- 结果 Re 中 Ri ($i=ID(v_2)$) 中所有元素 v_1 , 就是与 v_2 节点关联的所有信息采集节点;
- (5)中改变 $w(e_i)$ 由于每次加入新信息采集模块时, 安全警戒模块的负载能力将发生改变, $C=C'$, C' 为改变后的负载能力。

信息分析模块与信息共享模块:

只涉及到权为 3 与 4 的顶点, 取出该类顶点, 形成一个子图, 成为共享子完全图, 信息分析模块与信息共享模块的关联关系转变为在共享子图中找到支撑树的问题, 该树只有一个权为 4 的顶点且为树根, 其他顶点权皆为 3, 根节点只连接出一条边。该问题可等价于找到所有 $d=3$ 的顶点的最小支撑树 T , $d=4$ 的顶点再与 T 中最近点相连。算法如下:

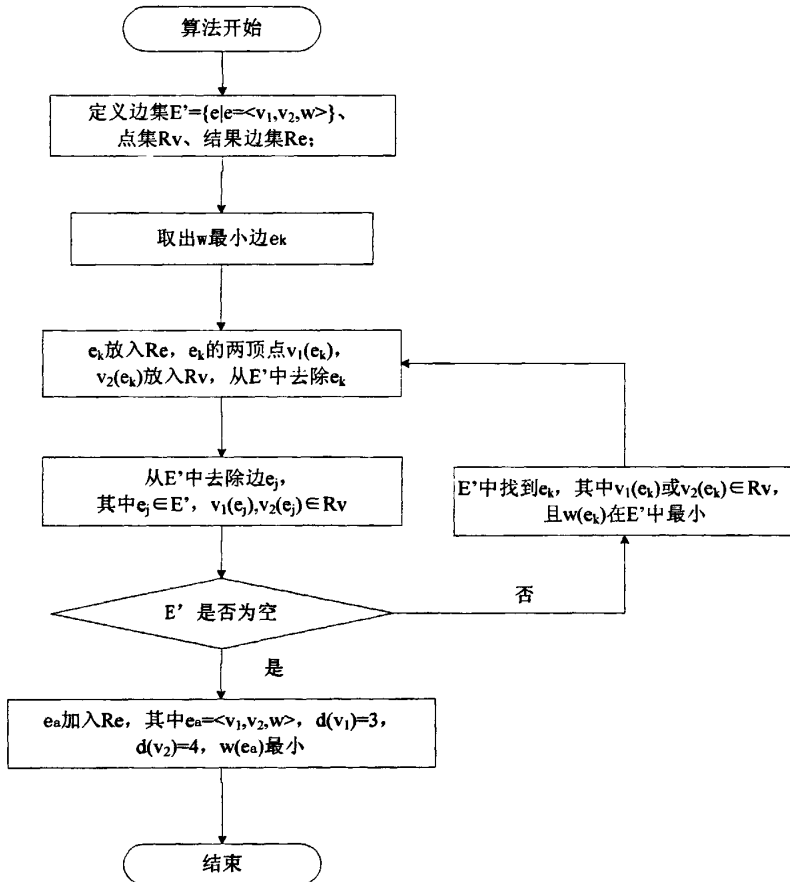


图 2.11 信息分析模块与信息共享模块负载均衡算法

- (1) 定义边集 $E' = \{e | e = \langle v_1, v_2, w \rangle\}$, 其中 $ID(v_1) \neq ID(v_2)$, $d(v_1) = d(v_2) = 3$ 为所有权为 3 的顶点间边的集合, 定义点集 R_v 、结果边集 R_e ;
 - (2) 从 E' 中找出 $w(e)$ 最小边 e_k ;
 - (3) 将 e_k 放入 R_e 中, e_k 的两顶点 $v_1(e_k)$, $v_2(e_k)$ 放入 R_v , 从 E' 中去除 e_k ;
 - (4) 将所有 $e_j \in E'$, e_j 两顶点 $v_1(e_j), v_2(e_j) \in R_v$ 的边 e_j 从 E' 中去除。若 $E' = \Phi$, 则将 $d=4$ 的顶点与所有 $d=3$ 的顶点相连的边中权最小的边加入 R_e , 算法结束; 若 $E' \neq \Phi$ 继续(5);
 - (5) 从 E' 中找到 e_k , 其中 $v_1(e_k)$ 或 $v_2(e_k) \in R_v$, 且 $w(e_k)$ 在 E' 中最小。重复(3)(4);
- 说明:
- 结果 R_e 组成的树就是共享子完全图的最小支撑树。

2.3.3 GOIDA 解决方案

上面章节分析了 GOIDS 在网格中的分布结构, 将 GOIDS 所在网格环境进行了模型化, 提出 GOIDA 负载均衡算法并针对 GOIDS 中的两种关联关系对算法进行了描述。下面将对 GOIDA 提出一种解决方案, 随后结合 GOIDA 解决方案, 以图 2.9 的网格结构为例, 对 GOIDS 负载均衡算法的执行过程进行讲解。

1. GOIDA 解决方案

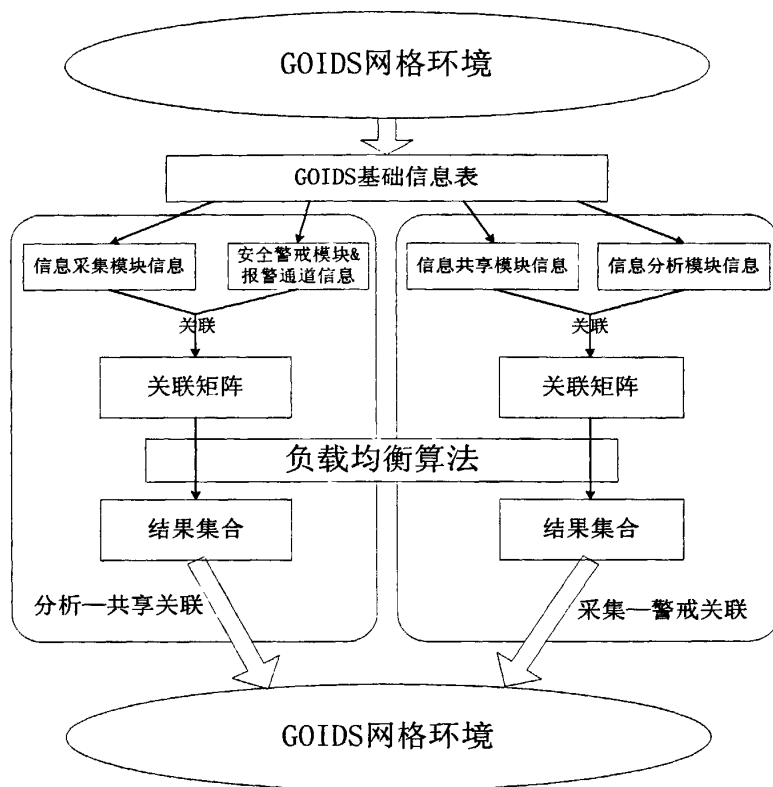
根据 GOIDS 模型的设计, GOIDA 解决方案分为网格中心控制节点与安全警戒模块两部分, 分别负责 GOIDA 对两类关联关系的维护与安全警戒模块信息采集模块之间的关联关系的缓存。

1) 网格中心控制节点

网格中心控制节点在网格环境中负责监控网格中各类资源的情况。在包含有 GOIDS 的网格环境中, 将对网格中心节点增加对 GOIDS 的监控功能, 包括:

- 维护基础模块信息表, 负责记录与维护网格环境中存在的所有 GOIDS 模块信息。
- 分别根据 GOIDS 中的两种关联生成两种关联矩阵, 通过 GOIDA 对安全警戒模块与信息采集模块之间的关联(以下简称警戒—采集关联)、信息分析模块与信息共享模块之间的关联(以下简称分析—共享关联), 进行计算与维护。
- 结果集合负责保存通过 GOIDA 计算得到的两种关联的结果, GOIDS 根

据结果集合中数据对网格环境中的各模块进行调度。



GOIDA 的解决方案如上图所示，图中最上方“GOIDS 网格环境”与最下方“GOIDS 网格环境”为同一环境。网格环境中所有的 GOIDS 模块信息都由基础模块信息表进行记录，根据 GOIDS 两种关联基础模块信息表中各类模块相关联得到关联矩阵，关联矩阵根据负载均衡算法得到结果集合，结果集合最终指导网格中 GOIDS 各模块的协同工作。采用这种方式实现可以使 GOIDA 的实现层次化，基础模块信息表负责 GOIDS 模块信息的记录，关联矩阵负责模块间关联关系的体现，结果集合负责指导 GOIDS 模块工作，通过关联关系基础模块信息表生成关联矩阵，通过负载均衡算法关联矩阵得到结果集合，各层次相互联系又相互独立，使 GOIDA 的实现更简单易行，同时各层次的可进行替代，例如不改变其他层次的情况下可以单独改变负载均衡算法，这为 GOIDA 的进一步研究与优化提供了方便，如文章第四章中负载均衡的测试中就采用了不同算法进行，

只需替换图中“负载均衡”那层就可实现。

下面将对 GOIDA 解决方案中的基础模块信息表、关联矩阵与结果集合以及相关的操作进行详细介绍。

(1) 基础模块信息表

网格中心节点维护一个基础模块信息表，该表中记录着 GOIDS 在网络环境中存在的所有 GOIDS 模块的信息，记录的信息包括：模块的唯一标识 (id)、所在节点的 ip 地址 (ip)、模块类型 (type)、初次加入网格时间 (faddtime)、上次加入网格时间(laddtime)、上次离开网格时间(lleavetime)、加入网格次数 (addcount)、模块有效标识 (isValid)，其中模块的唯一标识 id 通过 ip 地址与模块类型确定，模块类型根据 GOIDA 中定义，1-4 分别对应四种模块。当 GOIDS 模块加入与离开网格环境时将进行如下操作：

- 模块加入网格环境：检查基础模块信息表中是否存在与其 ip 及模块类型相匹配的记录，若存在对应记录则修改该记录中 laddtime 为此次加入时间，addcount 加 1，isValid 设为 1；若不存在匹配记录，则添加新记录，记录 ip、type、生成唯一标识 id、faddtime 与 laddtime 设为本次加入时间、lleavetime 与 addcount 为 0、isValid 为 1。
- 模块离开网格环境：模块离开网格环境是通过网格中心节点通过访问模块所在节点的网络速度探测接口实现的，当网络速度探测接口返回“节点无法连通”信息则认为该模块离开网格环境，网络速度探测接口的实现将在第三章第二节“系统构架与实现”中作介绍。当模块离开网格环境时，将基础模块信息表中该 ip 节点对应的所有记录的 lleavetime 设为当前时间，将 isValid 设为 0。

(2) 关联矩阵

GOIDS 基础模块信息表中对网格环境中的 GOIDS 功能模块进行了记录与维护，下面提出关联矩阵的方法对 GOIDS 的两种关联关系进行计算与维护，并将结果存储到结果集合中。

警戒—采集关联矩阵：

- 关联矩阵的生成：从基础模块信息表中找出所有 type=1、isValid=1 的模块，即所有信息采集模块，只需提出该模块的唯一标识，取出的节点记为节点向量 $V_I = \langle v_{I1}, v_{I2}, v_{I3}, \dots, v_{Im} \rangle$ ，其中共有 m 个信息采集模块， v_{Ii} 用模块的唯一标识 id 进行表示；同理从基础模块信息表中取出所有

安全警戒模块形成节点向量 $V_d = \langle v_{d1}, v_{d2}, v_{d3}, \dots, v_{dn} \rangle$ ，其中共有 n 个元素。生成矩阵：

$$A_{m \times n} = \begin{Bmatrix} W_{11} & W_{12} & \dots & W_{1n} \\ W_{21} & W_{22} & \dots & W_{2n} \\ \dots & \dots & \dots & \dots \\ W_{m1} & W_{m2} & \dots & W_{mn} \end{Bmatrix}$$

其中行 i 代表了第 i 个信息采集模块和所有安全警戒模块的关系情况，列 j 代表了第 j 个安全警戒模块的所有信息采集模块的关系情况， w_{ij} 代表第 i 个信息采集模块与第 j 个安全警戒模块之间的关系，也就是 GOIDA 中两节点间的边权 w 。

- 模块 k 加入：当网络环境中加入新信息采集模块 k 时，在基础模块信息表中添加了该模块的一条记录或将该模块对应的记录 $isValid$ 设置为 1，在 $V_I = \langle v_{I1}, v_{I2}, v_{I3}, \dots, v_{Im} \rangle$ 相应的加入一节点 v_{Ik} ， $A_{m \times n}$ 加入新行 k ，并采用访问 k 模块所在节点的网络速度探测接口测量 k 节点到所有 $V_d = \langle v_{d1}, v_{d2}, v_{d3}, \dots, v_{dn} \rangle$ 节点的网速值，并结合 V_d 中节点 L （逻辑关系）与 C （负载能力）设置 k 行中 w_{kj} 的值；

当加入新的安全警戒模块时，与信息采集模块加入时类似，不同之处在于在基础模块信息表中做添加安全警戒模块的记录，在 $V_d = \langle v_{d1}, v_{d2}, v_{d3}, \dots, v_{dn} \rangle$ 加入新节点 k ，在 $A_{m \times n}$ 加入新列 k ，并采用访问 k 模块所在节点的网络速度探测接口测量 k 节点到所有 $V_I = \langle v_{I1}, v_{I2}, v_{I3}, \dots, v_{Im} \rangle$ 节点的网速值，最终结合新节点 v_k 的 L 与 C 设置 k 列中 w_{ik} 的值。

- 模块退出：基础模块信息表中修改相应记录，包括 $leaveTime$ 、 $isValid$ ，在对应类型的节点向量 V_I 、 V_d 中删除该节点，并在 $A_{m \times n}$ 中删除对应行或列。
- 矩阵维护：采用触发与定期结合的方式对 $A_{m \times n}$ 进行维护，当有新模块加入网络环境时将采用模块加入的机制对新模块在 $A_{m \times n}$ 中对应的行或列进行更新；当一段时间没有发生模块加入事件时，则定期对 $A_{m \times n}$ 中元素进行更新，由于逻辑关系 L 相对比较稳定，且难以通过负载均衡

算法探知，所以不做更新。更新工作采集 $V_d = \langle v_{d1}, v_{d2}, v_{d3}, \dots, v_{dn} \rangle$ 中节点的 C ，以及通过逐个访问 $V_l = \langle v_{l1}, v_{l2}, v_{l3}, \dots, v_{lm} \rangle$ 中节点的网络速度测试接口对 $V_d = \langle v_{d1}, v_{d2}, v_{d3}, \dots, v_{dn} \rangle$ 进行测试，得到 N ，从而更新 $A_{m \times n}$ 中的 w_{ij} 。

分析—共享关联矩阵：

- 关联矩阵的生成：从基础模块信息表中找出所有 $type=3$ 、 $isValid=1$ 的模块，即所有信息共享模块，取出该模块的唯一标识，取出的节点记为节点向量 $V_S = \langle v_{S1}, v_{S2}, v_{S3}, \dots, v_{Sm} \rangle$ ，其中共有 m 个信息共享模块， v_{Si} 用的模块的唯一标识 id 表示。生成矩阵 $B_{m \times m}$ ：

$$B_{m \times m} = \begin{Bmatrix} W_{11} & W_{12} & \dots & W_{1m} \\ W_{21} & W_{22} & \dots & W_{2m} \\ & \dots & \dots & \\ W_{m1} & W_{m2} & \dots & W_{mm} \end{Bmatrix}$$

其中 w_{ij} 表示第 i 与 j 个信息共享模块之间的边权，由 GOIDA 中对网络环境的结构定义可知， $w_{ij}=w_{ji}$ ，当 $i=j$ 时， $w_{ij}=0$ 。

从基础模块信息表里找出所有 $type=4$ 、 $isValid=1$ 的模块，即所有信息分析模块，记节点向量为 $V_A = \langle v_{A1}, v_{A2}, v_{A3}, \dots, v_{An} \rangle$ ，共 n 个信息分析模块，结合 $V_S = \langle v_{S1}, v_{S2}, v_{S3}, \dots, v_{Sm} \rangle$ 。生成矩阵 $B'_{m \times n}$ ：

$$B'_{m \times n} = \begin{Bmatrix} W_{11} & W_{12} & \dots & W_{1n} \\ W_{21} & W_{22} & \dots & W_{2n} \\ & \dots & \dots & \\ W_{m1} & W_{m2} & \dots & W_{mn} \end{Bmatrix}$$

W_{ij} 表示用第 i 个信息共享模块与第 j 个信息分析模块之间的关系。

- 模块 k 加入：当加入的模块 k 为信息共享模块时，在基础模块信息表中做相应的记录，在 $V_S = \langle v_{S1}, v_{S2}, v_{S3}, \dots, v_{Sm} \rangle$ 加入相应的节点，在 B 中

加入对应的行与列， k 节点通过网格速度测试接口访问 $V_S = \langle v_{S1}, v_{S2}, v_{S3}, \dots, v_{Sm} \rangle$ 中节点得到 k 与 V_S 中节点间的网速值 N ，结合 L 与 C 得到 B 中 w_{ij} 值。在 B' 加入对应的行，通过网格速度测试接口 k 节点访问 $V_A = \langle v_{A1}, v_{A2}, v_{A3}, \dots, v_{An} \rangle$ 中节点得到 N ，结合 L 与 C 得到 B' 中 w_{ij} 值。

- 模块退出：基础模块信息表中修改相应记录，包括 `lleaveTime`、`isValid`，在对应类型的节点向量 V_S 、 V_A 中删除该节点，并在 B 与 B' 中删除对应行或列。
- 矩阵维护：触发与定期结合的方式对 B 与 B' 进行维护，当有新模块加入网格环境时将采用模块加入的机制对新模块在 B 与 B' 中对应的行或列进行更新；当一段时间没有发生模块加入事件时，则定期对 B 与 B' 中元素进行更新，采用 $V_S = \langle v_{S1}, v_{S2}, v_{S3}, \dots, v_{Sm} \rangle$ 中节点相互访问网络速度测试接口对 B 进行维护；矩阵 B' 通过逐个访问 $V_A = \langle v_{A1}, v_{A2}, v_{A3}, \dots, v_{An} \rangle$ 中节点的网络速度测试接口对 $V_S = \langle v_{S1}, v_{S2}, v_{S3}, \dots, v_{Sm} \rangle$ 的节点进行测试，得到相应的 N ，从而更新 B' 中的 w_{ij} 。

(3) 结果集合

结果集合负责保存 GOIDA 针对警戒—采集关联及分析—共享关联等到的结果，分为警戒—采集结果集合与分析—共享结果集合。

警戒—采集结果集合：

结果集合中保存有信息采集模块与安全警戒模块的关联关系，形式为 $Re_A = \{ \langle v_{11}, v_{21}, w_1 \rangle, \langle v_{12}, v_{22}, w_2 \rangle, \langle v_{13}, v_{23}, w_3 \rangle, \dots \}$ ，其中 v_{1i} 信息采集模块， v_{2i} 为安全警戒模块， $0 < i \leq n$ ， n 为信息采集模块数量， w 为警戒—采集关联关系的边权， Re_A 共包含 n 个信息采集模块与安全警戒模块的对应关系。

分析—共享结果集合：

结果集合中保存有信息共享模块间及与信息分析模块的关联关系，形式为 $Re_B = \{ \langle v_{31}, v_{32} \rangle, \langle v_{32}, v_{33} \rangle, \langle v_{31}, v_{41} \rangle, \dots \}$ ，其中 v_{3i} 信息共享模块， v_{4j} 为信息分析模块， $0 < i \leq n$ ， n 为信息共享模块数量， $0 < j \leq m$ ， m 为信息分析模块数量， Re_B 中共包含有 $n-1+m$ 个对应关系。

2) 安全警戒模块与报警通道

安全警戒模块与报警通道通过将警戒—采集结果集合中与本节点相关的关

联关系缓存到报警通道中，安全警戒模块根据报警通道中缓存的关联关系对信息采集模块进行检测。缓存内容为信息采集模块的 ip 地址信息，安全警戒模块根据 ip 地址访问相应信息采集模块的检测数据接口获得检测数据。接口的实现在第三章第二节中将有说明。

2. GOIDA 执行过程

1) 警戒—采集关联

以图 2.9 的网格环境中警戒—采集关联为例，下图是其检测子图，图中节点标记规则为“类型号+模块标号”，比如“21”代表类型为 2 即安全警戒模块，所在模块标号为 1 的模块的顶点，“112”代表类型为 1 即信息采集模块，模块标号为 12。图中右侧是该子图对应的警戒—采集关联矩阵，横轴代表安全警戒模块顶点，共 3 个，纵轴代表信息采集模块顶点，共有 12 个，矩阵中元素为安全警戒模块顶点与信息采集模块顶点间边权，即连通关系。本例中边权大小如图中关联矩阵所示，并规定安全警戒模块每增加一个关联的信息采集模块其相关的边权增加 0.2。

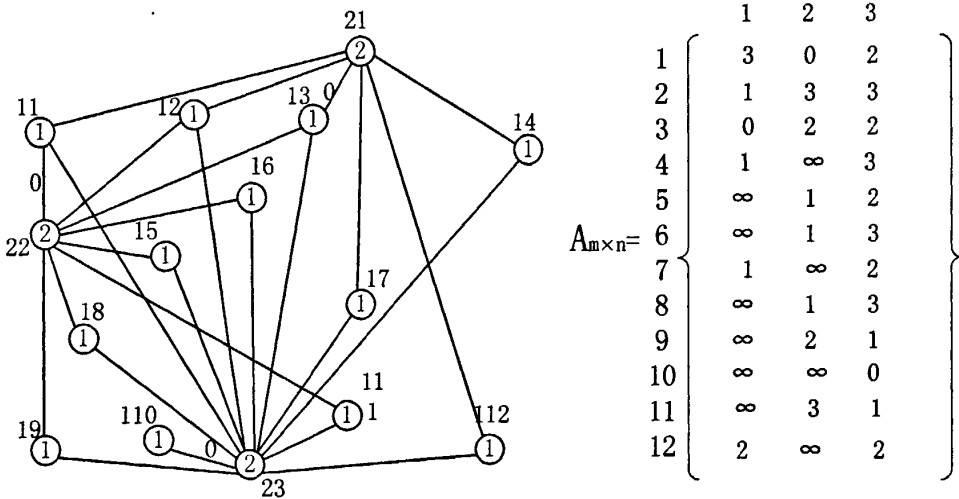


图 2.13 警戒—采集关联

下表中演示了负载均衡的运算过程及结果集合的情况，网格环境共有 12 个信息采集模块，所以算法共执行 12 步，每一步确定一个警戒—采集关联关系，本例中取初始状态、执行前三步与执行最后两步对算法进行介绍，下表为执行状态，其中 E' 为运算边集，Re 代表结果结合，E' 中黑体字标记为上一步改变边

权的边。

表 2.1 警戒—采集关联运算过程

步数		
0	E'	{<1,1,3>,<1,2,0>,<1,3,2>,<2,1,1>,<2,2,3>, <2,3,3>, <3,1,0>, ……<12,3,2>}
	Re	Re= Φ
1	E'	{<1,1,3>, <1,3,2>, <2,1,1>, <2,2,3>, <2,3,3>, <3,1,0>, ……<12,3,2>}
	Re	Re={R ₂ } R ₂ ={<1,2,0>}
2	E'	{<2,1,1>, <2,2,3.2>, <2,3,3>, <4,1,1>, <4,3,3>, <5,2,1.2>, ……<12,3,2>}
	Re	Re={R ₂ , R ₁ } R ₂ ={<1,2,0>} R ₁ ={<3,1,0>}
3	E'	{<2,1,1.2>, <2,2,3.2>, <2,3,3>, <4,1,1.2>, <4,3,3>, <5,2,1.2>, ……<12,3,2>}
	Re	Re={R ₂ , R ₁ , R ₃ } R ₂ ={<1,2,0>} R ₁ ={<3,1,0>} R ₃ ={<10,3,0>}
4-10		………
11	E'	{<12,1,2.8>, <12,3,2.6>}
	Re	Re={R ₂ , R ₁ , R ₃ } R ₂ ={<1,2,0>, <6,2,1.2>, <5,2,1.4> <8,2,1.6>} R ₁ ={<3,1,0>, <3,1,1.2>, <3,1,1.4>, <3,1,1.6>} R ₃ ={<10,3,0>, <3,1,1.2>, <3,1,1.4>}
12	E'	Φ
	Re	Re={R ₂ , R ₁ , R ₃ } R ₂ ={<1,2,0>, <6,2,1.2>, <5,2,1.4> <8,2,1.6>} R ₁ ={<3,1,0>, <4,1,1.2>, <2,1,1.4>, <7,1,1.6>} R ₃ ={<10,3,0>, <11,3,1.2>, <9,3,1.4>, <12,3,2.6>}

初始状态时边集 E' 中包含关联矩阵中除去无法连通的边（矩阵中标记为“∞”）之外的所有边，此时结果集 Re 为空。第一步选出边权最小边<1,2,0>，

在结果集中创建 R_2 ，并将 $\langle 1, 2, 0 \rangle$ 放入 R_2 中，从 E' 中除去所有信息采集模块编号为 1 的边，并更新所有安全警戒模块编号为 2 的边，进入第二步；第二步选取此时 E' 中权最小的边 $\langle 3, 1, 0 \rangle$ ，创建 R_1 ，将该边加入 R_1 ，并去除所有信息采集模块编号为 3 的边，将安全警戒模块编号为 1 的边的边权更新；重复上述操作直到 E' 中不再包含边，此时 R_e 中记录三个安全警戒模块与其相对应的信息采集模块的边，分别为编号为 2 的安全警戒模块对应 1、6、5、8 号信息采集模块，1 号安全警戒模块对应 3、4、2、7 号信息采集模块，3 号安全警戒模块对应 10、11、9、12 号信息采集模块。通过 GOIDA 的处理，图 2.9 就变为下图的结构，其中连线就是通过 GOIDA 对警戒—采集关联进行演算得到的结果。

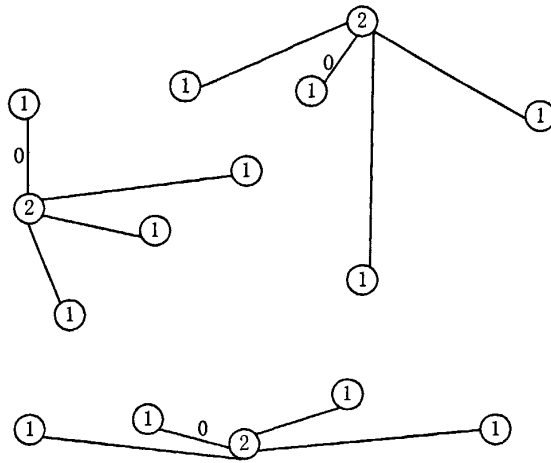


图 2.14 警戒—采集关联结果

得到关联结果后网格中心控制节点通过基础模块信息表得到各编号信息采集模块的信息，分别将其 ip 地址根据 GOIDA 结果发送至对应的安全警戒模块与报警通道的缓存中，安全警戒模块通过缓存中的 ip 地址对信息采集模块进行检测。

2) 分析—共享关联

从图 2.9 的网格环境中取出所有信息共享与信息分析模块节点，形成共享子图，并形成分析—共享关联矩阵，如下图所示，编号规则同警戒—采集关联示例，其中信息分析模块开头数字为 4，信息共享模块开头数字为 3。

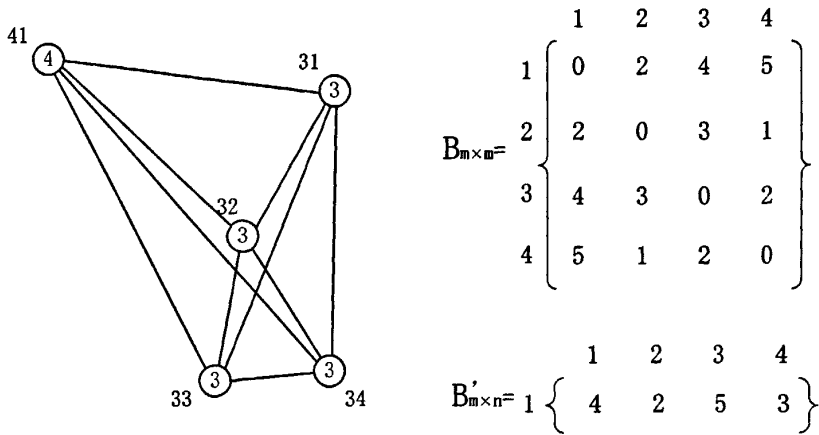


图 2.15 分析-共享关联

表 2.2 分析-共享关联运算过程

步数		
0	E'	{<1,2,2>,<1,3,4>,<1,4,5>,<2,1,2>,<2,3,3>,<2,4,1>,...<4,3,2>}
	Re	Φ
	Rv	Φ
1	E'	{<1,2,2>,<1,3,4>,<1,4,5>,<2,1,2>,<2,3,3>,<2,4,1>,...<4,3,2>}
	Re	{<2,4,1>}
	Rv	{2, 4}
2	E'	{<1,2,2>,<1,3,4>,<1,4,5>,<2,1,2>,<2,3,3>,...<4,3,2>}
	Re	{<2,4,1>,<1,2,2>}
	Rv	{2, 4, 1}
3	E'	{<1,3,4>,<2,3,3>,<3,1,4>,<3,2,3>,<3,4,2>,<4,3,2>}
	Re	{<2,4,1>,<1,2,2>,<3,4,2>}
	Rv	{2, 4, 1, 3}
4	E'	Φ
	Re	{<2,4,1>,<1,2,2>,<3,4,2>} R'e={<2,1,1>}
	Rv	{2, 4, 1, 3}

说明：R'e 为关联矩阵 B' 中信息分析模块与信息共享模块关联权最小边。

初始状态时边集 E' 中包含关联矩阵中除去无法连通的边（矩阵中标记为“ ∞ ”）之外的所有边，结果边集 Re 与点集 Rv 为空。第一步选出边权最小边 $\langle 2,4,1 \rangle$ ，在 Re 中记录此边，将此边两顶点 2 与 4 加入 Rv 中，从 E' 去除所有两顶点均在 Rv 的边，进入第二步；第二步选取边权最小边，该边要有一个节点在点集 Rv 中，此时选出边为 $\langle 1,2,2 \rangle$ ，加入 Re ，在 Rv 中加入 1，此时 $Rv=\{2,4,1\}$ ，从 E' 去除所有两顶点均在 Rv 的边，进入下一步；最终 Rv 中包含了所有初始 E' 中的所有结点， E' 为空， Re 中为信息共享模块间最终的结果边集，再从 B' 矩阵中取出信息分析模块与信息共享模块间边权最小的边，得到分析共享一关联的结果集。

通过 GOIDA 的处理，图 2.9 就变为下图的结构，其中细线代表安全警戒模块与信息采集模块的关联，粗线代表了信息分析模块与信息共享模块的关联

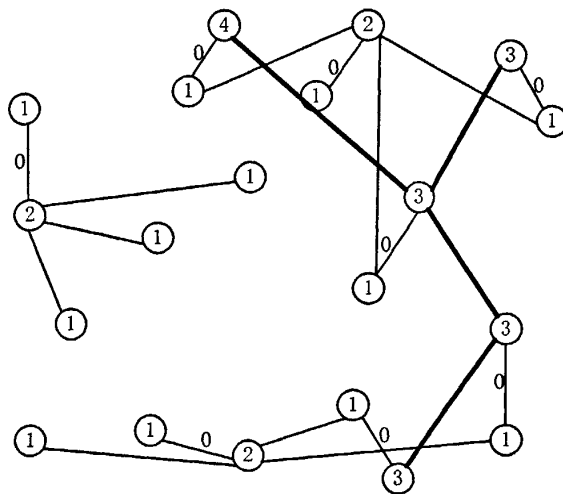


图 2.16 算法结果

本文第四章中将 GOIDS 部署于天津市郊区工业信息网的网格环境中，并根据该网格特点对 GOIDA 各参数进行了设定。以该网格环境为基础，对 GOIDA 进行了测试，测试结果表明通过 GOIDA 的调度 GOIDS 在网格中具有较好负载均衡效果。

第三章 应用技术与系统实现

GOIDS 的实现使用到开源软件 Snort 及 Web 服务与 XML 技术,本章将结合这些应用到的技术进行介绍,并随后对系统中关键点的实现进行讲解,包括系统构架、接口与上章中 GOIDA 解决方案的实现。

第一节 应用技术介绍

GOIDS 中采用开源软件 Snort 的嗅探器作为信息采集模块中网络数据采集器,安全警戒模块中使用 Snort 的入侵检测器作为数据检测器,这样在保障入侵检测能力的同时可以有效减少 GOIDS 的开发成本,一些已提出的系统也采用这类实现方式,比如 R-GMA^[10] 采用 Snort 作为其入侵检测的核心模块。

由于 Web 服务与 XML 具有跨平台、跨防火墙通讯等特点,正满足网格环境中 GOIDS 模块传输数据的要求,在 GOIDS 的实现中,接口与模块间的传输分别使用了 Web 服务与 XML 技术。

下面结合在 GOIDS 实现中的应用对 Snort 及 Web 服务与 XML 技术分别进行介绍。

3.1.1 Snort 在 GOIDS 中的应用

Snort^[21]是一款由 SourceFire 组织开发的轻量级开源网络入侵检测系统软件,它使用规则描述语言,提供了实时数据流量分析、网络数据包记录与数据包日志文件检测的功能。它能够进行协议分析,对数据内容进行搜索与匹配,检测各种不同的攻击方式,对攻击进行实时报警。同时 Snort 遵从通用公共许可证 GPL(General Public License),任何一个遵守 GPL 的组织或个人都可以自由是用该软件。

1. Snort 特点:

- Snort 是一款轻量级入侵检测系统软件,代码简洁,完全程序安装包文件不到 3MB,如果只使用嗅探器安装程序将更小;
- 现已支持 Linux、Unix、Solaris、BSD、Windows 等多种操作系统,具有

极好的可移植性，这为基于网络入侵检测系统的实现提供了保障；

- 具有良好的扩展性，它使用一种简单的规则描述语言，这种描述语言非常易于扩展，通过简单编写就可为 Snort 添加新的检测规则，并且 Snort 支持插件，通过外接插件就可以增加特定的检测功能；
- 遵从通用公共许可证 GPL(General Public License)，任何一个遵守 GPL 的组织或个人都可以自由是用该软件，包括免费使用并且可以自由更改代码添加在其他应用程序中。

2. Snort 工作模式

Snort 有三种工作模式：嗅探器模式，分组日志模式，网络入侵检测模式。嗅探器模式实现从网络上读取数据包，并以数据流的形式显示在终端上；分组日志模式把网络上读取的数据包记录到硬盘的日志文件中；网络入侵检测模式分析网络中传输的数据并与入侵规则库相匹配来发现入侵行为。

- 嗅探器模式 (Sniffer Mode):

Snort 使用 Libpcap 或 Winpcap 包捕捉库。在这种模式下，Snort 使用指定的网络接口在混杂模式下读取并解析网络中的分组报文。常用指令：

```
snort -i x -v -d -e
```

将把数据包的包头、应用层数据信息输出到终端。

参数说明：

-i x: 在网络接口 x 上监听网络数据

-v: 显示 TCP/IP 数据报头信息

-d: 显示应用层数据

-e: 显示并记录第二层信息包头的的数据

GOIDS 中应用：

嗅探器模式是 Snort 的最小模式，只需 Snort 中三个文件就可运行，仅提供网络数据包抓取工作，正适合信息采集模块中使用，信息采集模块使用该模式对学习数据采集进行捕捉，命令如下：

```
snort -i 2 -v -d -e
```

表示在第二个网络接口监听数据，包括 TCP/IP 包头信息、应用层信息、第二层信息报头数据，将 Snort 返回的结果数据流存入缓存中，经过格式化处理后通过信息学习接口提交到信息共享模块。

- 分组日志模式 (packet logger Mode):

在这种模式下，需要指定记录包的目录，常用指令：

```
snort -i x -d -e -v -l ./log
```

它将把网络上的数据包记录到./log 所指定的目录中。

参数说明：

-l : 记录网络数据包到指定目录的日志文件中

● 网络入侵检测模式 (network intrusion detection system Mode):

Snort 最重要的模式，作为网络入侵检测系统(NIDS)，常用指令：

```
snort -d -h 192.168.1.0/24 -l ./log -c snort.conf
```

此时 Snort 载入 snort.conf 配置文件，将 192.168.1.0/24 此网络的报警信息记录到./log 中去。snort.conf 文件可以换成用户自己的配置文件，载入 snort.conf 配置文件后 snort 将会根据 snort.conf 中的规则去判定每一个数据包是否合法。

参数说明：

-h : 设置内网地址，设置此选项输出结果会使用箭头表示数据进出的方向

-c : 将数据与规则文件中规则匹配，实现入侵检测功能

GOIDS 中应用：

GOIDS 中安全警戒模块结合分组日志模式与网络入侵检测模式对信息采集模块提交的数据进行检测，命令如下：

```
snort -i 2 -dver C:\Snort\log\snort.log -c c:\snort\etc\snort.conf  
-l c:\snort\log
```

此时 Snort 将根据 snort.conf 中的配置信息选取规则库，将指定文件中的数据进行检测，当发现入侵行为时返回入侵警报并记录在日志文件中，GOIDS 安全警戒模块将信息采集模块提交的信息存储于临时文件中使用上述参数设置进行监测。

3. Snort 报警模式：

Snort 在网络入侵检测模式下有 5 种报警机制：fast, full, console, syslog, test 和 none。

常用的三种模式分别是：

fast: 报警信息包括：一个时间戳、报警消息、源/目的 IP 地址和端口；

full: 默认的报警模式；

none: 关闭报警机制。

这三种模式都通过参数-A, 进行设置, syslog 模式使用参数 M。

GOIDS 中应用:

还有一种模式不属于报警模式, 但会经常使用到, 就是使用 Tcpdump 格式记录 LOG 的信息包, 使用参数 b 设置, 它将所有信息包都记录为二进制形式, 用这个选项不需要把信息转化为文本的时间, 所以记录速度相对较快。命令为:

```
snort -i 2 -b -A fast -l c:\snort\log
```

采用这种方式 Snort 将返回网络信息的二进制数据流, 该命令可运行在嗅探器模式下, GOIDS 信息采集模块通过上述命令得到网络数据流, 并转换为 XML 文件通过检测数据接口提交给安全警戒模块进行检测。

4. 规则改写

在信息分析模块发现新类型攻击时会向安全警戒模块发送安全规则更新。使用的方法是通过安全警戒模块上的数据接收接口远程在 Snort 的检测规则文件 local.rules 中写入规则, 规则形式为:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP";  
dsize:>512; itype:8; reference:arachnids,162; classtype:bad-unknown;  
sid:499; rev:3;)
```

该段规则的意思是: 检测 ICMP 包, 发现来自 \$EXTERNAL_NET 被定义为 (default = any) 或 \$HOME_NET 被定义为 (default=any), 数据大小 (dsize) 大于 512 字节, ICMP (itype) 类型是 8 的数据包, 就发出告警。

信息采集模块根据新类型攻击的特点编写规则, 并发送到安全警戒模块进行更新。

3.1.2 XML 与 Web Service 在 GOIDS 中的应用

GOIDS 模块存在于网格节点中, 网格节点的操作系统各不相同, 这就使模块间的数据传输数据传输涉及到跨平台的问题, 针对这一问题 GOIDS 采用 XML 技术数据传输, 接口使用 Web 服务进行了实现。

1、XML 在 GOIDS 中的应用

XML^[22] (eXtensible Markup Language 可扩展标记语言) 是由 W3C^[22] (World Wide Web Consortium) 定义的一种标记语言, 它通过使用有意义的结构对信息进行编码, 使计算机和人在一定程度上可以理解这种编码。XML 文档组成: 一个

格式正规的 XML 文档由三个部分组成：一个可选的序言；文档主体；可选的尾声。在本文中，GOIDS 信息采集模块中学习数据采集接口返回的是一段 XML 文档代码，下例中用注释的方式对这段代码进行了解释，注释为“<!--.....>”标记信息。

```

<?xml version="1.0" encoding="utf-8" ?><!--序言-->
<!--文档主体开始-->
<string xmlns="http://Localhost/"><!--根元素，元素名为 string，属性值为
                               xmlns=http://Localhost/-->
    <SnortData><!--根元素一级子元素-->
        <Packet><!--根元素二级子元素，SnortData 一级子元素-->
            <time>2008/4/8-15:33:30.322304</time><!--2008/4/8
            -15:33:30.322304 为元素 time 的字符数据-->
            <type>ARP</type>
            <src_addr>192.168.2.1</src_addr>
            <src_port /><!--空元素标记-->
            <des_addr>192.168.2.222</des_addr>
            <des_prot />
            .....
        </Packet>
    </SnortData>
</string>
<!--文档主体结束-->
<!--comments and processing instructions allow here --><!--尾声-->

```

2、Web 服务技术在 GOIDS 中的应用

Web 服务与传统 Web 应用程序相比具有以下优势：跨平台、跨防火墙通讯、用程序集成简单、B2B 集成简单、软件与数据复用性高等。下面以 GOIDS 信息采集模块的学习数据采集接口为例，采用 Web 方式在 WebService 文件.asmx 中添加以下代码：

```

[WebMethod(Description = "学习数据接口")]//声明为Web服务方式
public string GetSnortData(string StopT)//参数为学习数据采集时间
{
    DataSet ds= new DataSet("SnortData");//声明数据集
    string SnortResultStr = Function.GetSnort(StopTi);//采集学习数据，返回一字符串
}

```

```
SnortResultStr = Function.FormatSnort(SnortResultStr); //对字符串进行格式化
return Function.dataout(SnortResultStr); //调用数据送出接口送出数据
}
```

上面的代码就是一个完整的 Web 服务，通过 WebMethod 声明将函数声明为 Web 服务形式，最终将采集到的学习数据通过数据送出接口封装为 XML 文件方式发送给信息共享模块，上一小节中“XML 文档组成”中的示例就是该函数返回的 XML。数据送出接口将在下一节做详细介绍。

客户端：

以下是 GOIDS 信息共享模块的信息接收接口的一段示意代码，共享模块采用 Winform 方式实现

```
private void getSnortData()
{
    string strURL = "http://" + tbIP.Text + "/WebS/InfoClt.aspx/GetSnortData?StopT=10";
    //根据设置，访问指定的信息采集模块的学习数据接口，tbIP.Text为采集模块地址
    string strValue = Function.dataout(strURL); //调用数据接收接口
    XmltoData(strValue); //将数据存入到数据库中
    Reader.Close();
}
```

通过数据接收接口获得指定位置的信息采集模块学习接口的数据，并存入到数据库中，数据接收接口将在下一节做详细介绍。

第二节 系统构架与实现

本文第二章从系统模型的角度对 GOIDS 的整体结构与各模块功能和设计进行了介绍，本节将从系统实现的角度给出 GOIDS 的系统构架，并对系统构架中的关键点——接口的实现进行详细说明。

3.2.1 GOIDS 系统构架与主要接口实现

文章到这里已经对 GOIDS 模型体系结构、各模块设计、开发应用的技术进行了介绍。下面将从 GOIDS 系统构架入手，对 GOIDS 中的接口进行介绍与分类，并对主要接口的实现作一个详细介绍。

1. GOIDS 系统构架

GOIDS 共包括五种模块，各模块通过各种类型的接口进行通信与数据传输，以达到协同工作的目的，完成 GOIDS 的面向网络环境的入侵检测。GOIDS 的系统构架如图 3.1 所示，其中标示了各个模块的接口，以及接口之间的通信方式。接口采用 Web 服务形式编写，数据传输采用 XML 方式，请求方式采用 HTTP-GET。

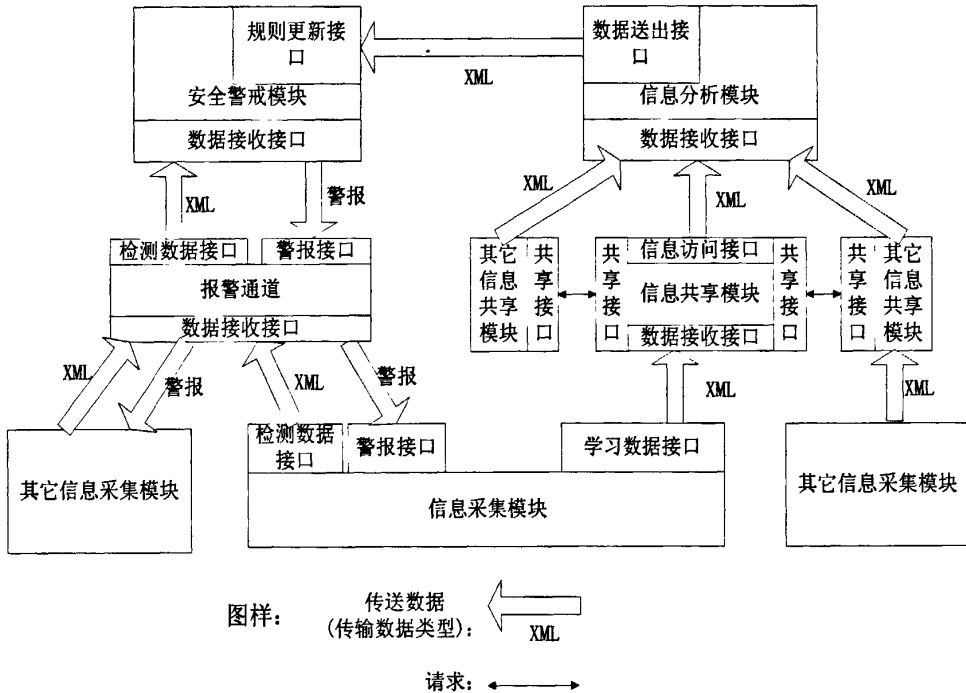


图 3.1 GOIDS 系统构架

所有接口分为两大类：共享接口，数据接口。

共享接口为信息共享模块中的共享接口，负责在信息共享模块之间通信，实现信息共享集的形成。共享接口间的通信请求以线箭头的形式表示。

数据接口为在图中除去信息模块共享模块共享接口外的其它所有接口，它负责数据的送出与接收，实现 GOIDS 中的数据传输工作。数据接口分为两种：数据送出接口、数据接收接口，分别负责数据的送出与接收工作。在图中数据接口传输数据用宽箭头形式表示，箭头方向表示数据流动方向。在系统设计与实现中为区分接口，各模块的接口名称不完全相同，但实际实现时只包含数据送出、数据接收的区别，在宽箭头所指方向接口为数据接收接口，宽箭头指出的方向为数据送出接口。

所有宽箭头两端的数据传输接口都可通过 HTTP-GET 方式进行请求。

2. 主要接口实现

1) 数据接口

数据接口分为：数据送出接口，数据接收接口。数据送出接口负责将节点中的数据进行处理，然后进行 XML 封装发送到其他节点中；数据接收接口负责接收其他节点发送来的 XML 文件并解析，根据节点需要格式化数据。

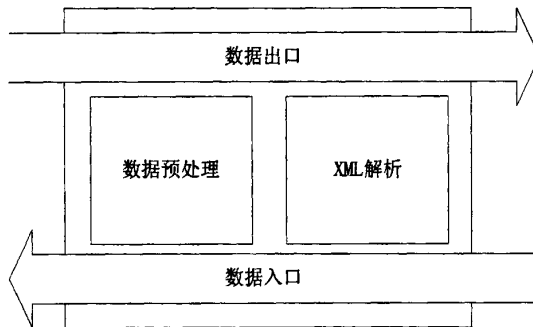


图 3.2 数据接口

数据送出接口实现代码如下，其中函数参数为一字符串，函数返回 XML 格式数据，函数体中依次进行参数读取、数据处理、XML 化处理、返回 XML 四个步骤。

```
public string dataout(string strIn) {  
    DataSet ds= new DataSet("dataset");//声明数据集  
    ds =Format(strIn);//将字符串格式化处理，存储在规则的数据集中  
    System. IO. StringWriter writer = new System. IO. StringWriter(strbuilder);  
    ds. WriteXml(writer, System. Data. XmlWriteMode. IgnoreSchema);//将数据集转为XML格式  
    return strbuilder. ToString();//返回XML格式学习数据  
}
```

数据接收接口实现代码如下，函数参数为要读取的数据送出接口Web服务地址，返回字符串数据。

```
public string datain(string strURL) //参数为要访问的数据送出接口的 Web 服务地址  
{  
    System. Net. HttpWebRequest request; //创建一个HTTP请求  
    request = (System. Net. HttpWebRequest)WebRequest. Create(strURL);  
    System. Net. HttpWebResponse response;  
    response = (System. Net. HttpWebResponse)request. GetResponse();  
}
```

```
System.IO.Stream s;  
s = response.GetResponseStream();//将HTTP响应存入到数据流中  
XmlTextReader Reader = new XmlTextReader(s);//将数据流转换为XML读取器  
Reader.MoveToContent();//跳至XML内容信息  
string strValue = Reader.ReadInnerXml();//读取XML内容信息  
Reader.Close();  
return strValue;  
}
```

这段代码实现了访问指定的地址的Web服务，得到XML格式数据，再将数据存入到数据库的过程。

2) 信息共享模块共享接口

GOIDS 的信息共享模块负责接收信息采集模块提供的学习数据，并为信息分析模块提供学习数据集。下图是信息分析模块获得学习数据集的流程，其中信息共享模块共享接口实现将请求参数传送给 GOIDA 中相邻的所有其他信息共享模块，相邻信息共享模块根据参数返回学习数据集并再次调用其相邻模块的过程，直到所有信息共享模块都向信息分析模块返回了学习数据集。

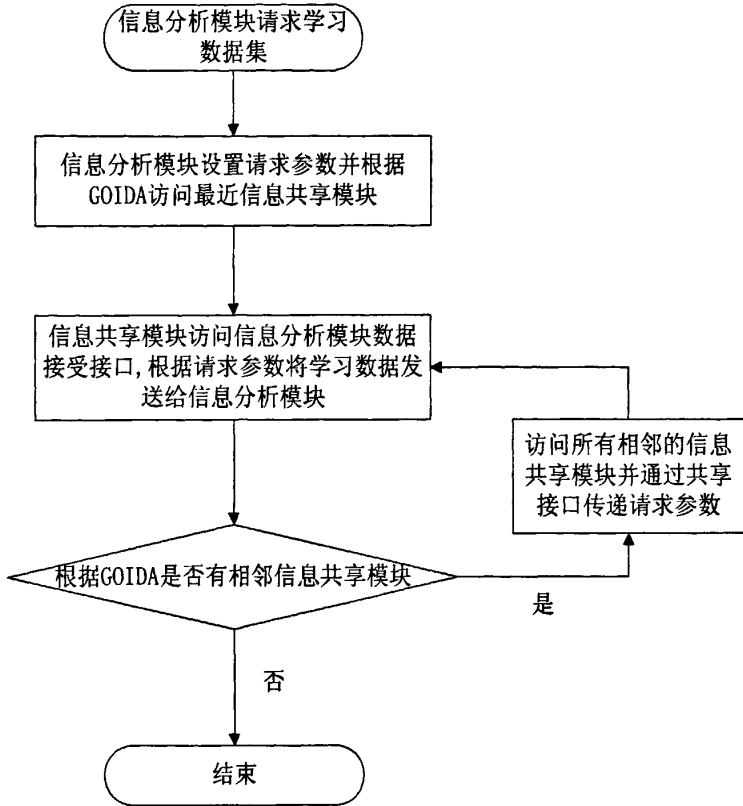


图 3.3 学习数据集获取流程

代码如下：

```

[WebMethod(Description = "共享接口")]//声明为Web服务方式
public void shareData(string srcShare, string ArgString)//参数为来源节点地址与请求参数
{
    Function.setSample(ArgString); //根据请求参数返回学习样本数据并发送至信息分析模块
    string strClosed = Function.getClosedShare(localip); //获取节点所有相邻信息共享模块
    strClosed = strClosed.remove(srcShare); //去处源请求地址, 防止产生回路
    Array ArrClosed = Function.StrToArr(strClosed); //地址字符串转化为数组
    foreach(string ip in ArrClosed)
    {
        shareData(ip, ArgString) //访问本节点所有相邻信息共享模块的共享接口
    }
}
    
```

3) 网络速度探测接口

网络速度探测接口分布于所有网格节点之中，该接口不属于 GOIDS 的任何模块，但对 GOIDA 的正常运行起着极其重要的作用，网格中心节点通过该接口探测接口所在节点与其他节点的网络连接速度状况，用来记录 GOIDA 中 $w = \langle N, L, C \rangle$ 中的 N 。

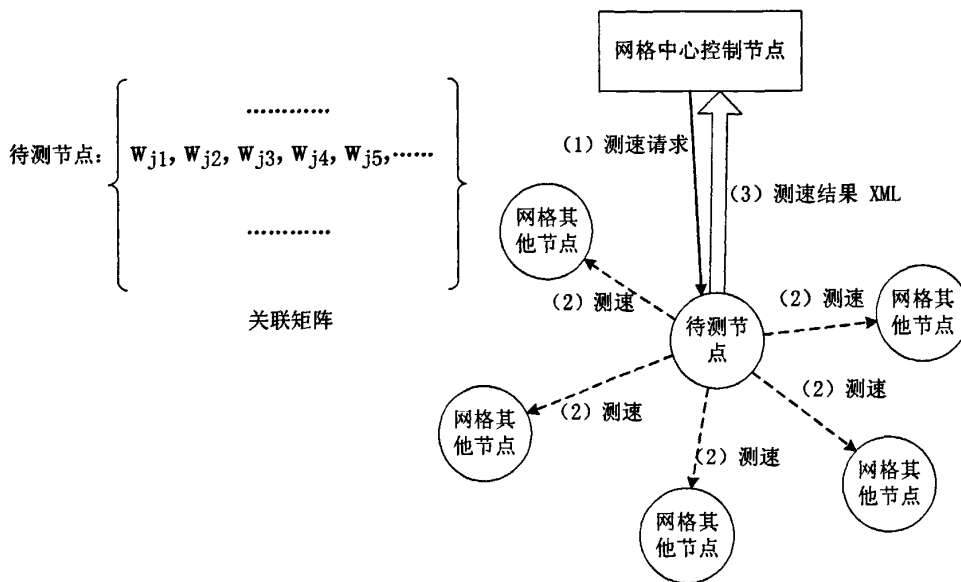


图 3.4 网络速度探测接口工作示意图

图 3.4 中对网络中心控制节点访问网路速度探测接口的过程与机制进行了解释。当网格中心节点中 GOIDA 需要维护关联矩阵中某一待测节点相关的边权 w_j 时，网格中心节点将关联矩阵中与待测节点相关网格节点的地址作为参数访问待测节点网络速度探测节点，即为图中 (1) 测速请求；测速节点根据参数访问其他网格节点并记录下网络连接数据，图中标记为 (2) 测速；得到测速结果后，待测节点将测速结果进行整理，并封装为 XML 格式，返回给网络中心控制节点，图中显示为 (3) 返回测试结果 XML，此时完成了网络中心控制节点对待测节点的访问过程。随后网络中心控制节点根据返回结果对关联矩阵中待测节点相关 w_j 进行维护。

网路速度探测接口代码为：

```
[WebMethod(Description = "网络测速接口")]//声明为Web服务方式
public string GetPing(string ip)//测试目标节点的地址
{
```

```

DataSet ds= new DataSet("NetSpeed");//声明数据集
String PingResultStr = Function.Ping(ip);//获得网络连接数据, 返回一字符串
PingResultStr = Function.FormatPing(PingResultStr);//格式化字符串
return Function.dataout(PingResultStr); //调用数据送出接口送出数据
}

```

网络连通时返回 XML 形式为:

```

<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="GOIDS">
  <NetSpeed><!--根元素一级子元素-->
    <222.129.11.36><!--探测目标节点的域名, 无域名则为 ip-->
      <ip>222.129.11.36</ip><!--探测目标节点 ip 地址-->
      <TTL>53</TTL><!--TTL 信息-->
      <Min>8</Min><!--最小回应时间-->
      <Max>13</Max><!--最大回应时间-->
      <Avg>10</Avg><!--平均回应时间-->
    </222.129.11.36>
  </NetSpeed>
</string>

```

网络中心节点获得上述 XML 后, 取 $N = (255 - TTL) \times 0.01 + Avg \times 0.09$, 获得该网络连接在 GOIDA 的网速值 N。

当网络无法连通时将返回:

```

<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="GOIDS">
  <NetSpeed>
    <222.129.11.0>
      0
    </222.129.11.0>
  </NetSpeed><!--节点无法连通该 ip-->
</string>

```

3.2.2 GOIDS 工作测试

在校园网中进行了如下测试: 配置一个包含 GOIDA 的节点, 并且该节点只包含信息采集模块 (以下称采集节点), 节点 ip 地址为 202.113.229.250, 该节点只包括数据提交与警报响应的功能。配置一包含安全警戒模块的节点 (以下简称安全警戒节点), 节点 ip 地址为 202.113.29.191。物理位置上这两个节点计算机分别在校园中的不同教学楼中, 在网络结构上它们相隔多层网络路由与交换

设备。将 202.113.29.191 添加至 202.113.229.250 的 GOIDA 基础模块信息表中，经过 GOIDA 负载均衡算法的计算得到 202.113.229.250 的信息采集模块与 202.113.29.191 的安全警戒模块相关联，并将关联结果通过数据送出接口发送到 202.113.29.191 的缓存文件中，202.113.29.191 中的缓存文件出现下图所示信息：



图 3.5 安全警戒模块运行效果图

这两个节点产生关联，随后在 202.113.29.191 安全警戒节点上运行的 GOIDS 安全警戒模块中出现如下图所示情况：

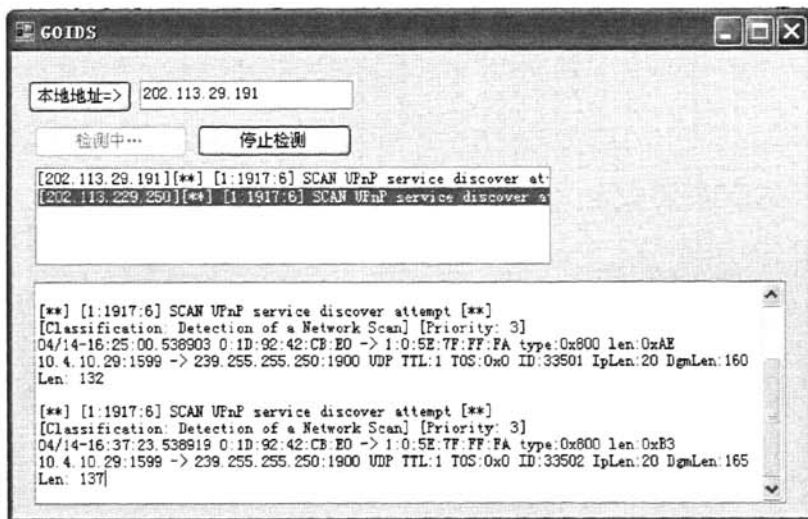
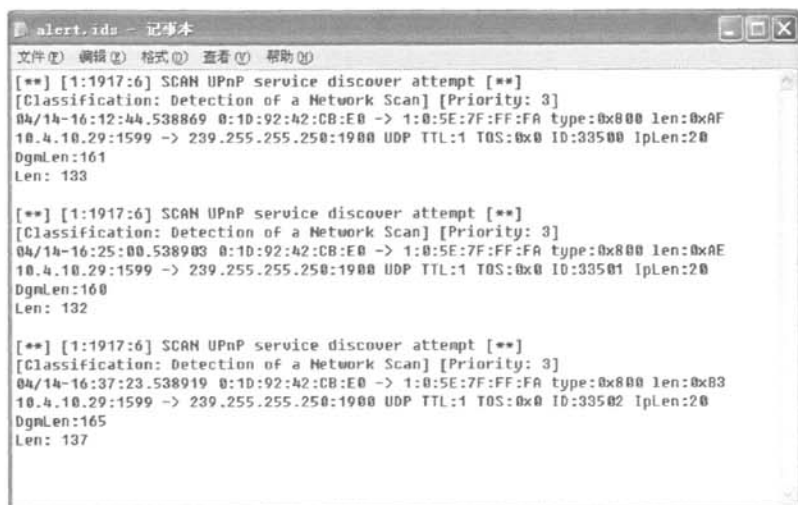


图 3.6 安全警戒模块运行效果图

其中该节点对本节点 202.113.29.191 与远程 202.113.229.250 两个节点的信息采集模块提交数据都进行了检测，在对 202.113.229.250 的检测中发现入侵行为，

入侵行为名称是 SCAN UPnP service discover attempt, 入侵级别为 3, 此攻击是 202.113.229.250 为网关的内网中地址为 10.4.10.29 广播的 UDP 数据包。安全警戒模块在发现入侵行为的同时将警报与入侵行为信息发送给检测数据的来源节点 202.113.229.250 中, 此时在 202.113.229.250 的入侵记录文件 alert.ids 中出现了如下信息:



```
alert.ids - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

[**] [1:1917:6] SCAN UPnP service discover attempt [**]
[Classification: Detection of a Network Scan] [Priority: 3]
04/14-16:12:44.538869 0:10:92:42:CB:E0 -> 1:0:5E:7F:FF:FA type:0x800 len:0xAF
10.4.10.29:1599 -> 239.255.255.250:1900 UDP TTL:1 TOS:0x0 ID:33500 IpLen:20
DgmLen:161
Len: 133

[**] [1:1917:6] SCAN UPnP service discover attempt [**]
[Classification: Detection of a Network Scan] [Priority: 3]
04/14-16:25:00.538903 0:10:92:42:CB:E0 -> 1:0:5E:7F:FF:FA type:0x800 len:0xAE
10.4.10.29:1599 -> 239.255.255.250:1900 UDP TTL:1 TOS:0x0 ID:33501 IpLen:20
DgmLen:160
Len: 132

[**] [1:1917:6] SCAN UPnP service discover attempt [**]
[Classification: Detection of a Network Scan] [Priority: 3]
04/14-16:37:23.538919 0:10:92:42:CB:E0 -> 1:0:5E:7F:FF:FA type:0x800 len:0xB3
10.4.10.29:1599 -> 239.255.255.250:1900 UDP TTL:1 TOS:0x0 ID:33502 IpLen:20
DgmLen:165
Len: 137
```

图 3.7 信息采集模块警报信息图

ip 为 202.113.229.250 采集节点只包括信息采集模块, 未包含任何安全检测机制。通过在 GOIDA 信息基础表中添加节点信息并通过负载均衡算法计算, 该采集节点找到一安全警戒节点与之相关联。安全警戒节点接收信息采集节点发送的检测数据进行检测。最终信息采集节点接收到来自非本地的安全警戒节点发送的警报与入侵行为信息。说明 GOIDS 对远程包含采集节点能进行有效的入侵检测, 并具有很好的扩展能力。

第四章 GOIDS 应用于天津市郊区工业信息网

《天津市郊区工业信息网》是科委、信息办支持的项目，论文作者参与该项目的系统构架与功能设计工作和部分功能的实现。在该网格环境中进行 GOIDS 系统实现与配置，具有以下优势：

- 论文作者参与了该项目的前期调研、系统构架、功能设计与实现工作，对该网格环境与服务对象——天津市郊区企业都较为了解，对网格节点、结构特点有较深的认识，有利于 GOIDA 各参数的设定。
- 作者作为该项目的主要开发者对网格环境中服务器的使用拥有一定便利条件，对网格环境具有较高级别的访问权限，为 GOIDS 的安装配置与后期的 GOIDA 测试提供了方便。

并且，在天津市郊区企业信息网中实现 GOIDS 具有以下意义：

- 天津市郊区工业信息网中的实施 GOIDS，增强了该网格环境的容错与入侵检测能力，提高了该网格的运行能力，从而达到了提高天津市郊区工业信息网的整体作业处理能力的目的，对天津市郊区工业的信息化建设做出了贡献。
- GOIDS 应用于天津市郊区工业信息网的网格环境，使网格技术与实际应用相结合；通过实际运行中的数据采集及运行状况分析，为网格的入侵检测系统的进一步研究提供了支撑。

本章首先对天津市郊区工业信息网的网格特点进行了分析，针对这些特点对 GOIDS 各模块与 GOIDA 参数进行了相应的配置，最后以天津市郊区工业信息网网格环境为背景对 GOIDS 检测能力与负载情况进行了测试与实验。

第一节 天津市郊区工业信息网的网格特点

天津市郊区工业信息网（以下简称：郊区工业网）以发展天津市郊区工业为宗旨。以建立一个市管理部门为中心、十二个区县服务器与服务网络为支撑、部分专业服务与专业网络为补充的服务网格平台为总体目标。目标在十一五期间实现天津市乡镇企业信息化服务与管理；为 1000 家乡镇企业提供网络交互平

台；为 3000 个乡镇企业提供数据、计算、信息等网格服务；使 4 万户中小企业获得信息咨询与数据共享；拥有 1 万户获得经常性服务的中小企业会员。

1、天津市郊区工业信息网的网格节点特点：

郊区工业网中存在的节点包括：市乡镇管理局、管理局职能部门、各区县管理局、区县管理局职能部门、乡镇企业，按照动态性可以分为三类：

- 1) 固定节点：市乡镇管理局，各区县管理局，该类节点作为主要管理部门长期存在于网格中，可作为网格的关键节点。
- 2) 动态性中等：区县管理局部分部门，市管理局部分部门，这类用户节点由于工作的需要，会出现一段时间内稳定存在于网格中，如进行年底总结需要收集乡镇企业数据，或者某段时间提供项目申报服务；又会出现一段时间离开网格，这是由于这些管理部门属于职能权力部门，保存有安全性资料，所以会在无工作需要时离开公网，也就离开了网格环境。
- 3) 动态性较强：乡镇企业，当需要使用网格资源或需要这类节点提供信息时接入网格，并可能使用完资源或提供完资源后就会离开网格，会出现短时间内多次接入并离开网格的情况，动态性最强。

2、网格结构特点：

郊区工业网以服务网格的形式实现，其中节点包括市区县各级管理局与部门以及乡镇企业，管理部门间、乡镇企业间以及管理部门与企业间存在有各种关系，这就使该网格在网络物理关系上还存在逻辑关系，其结构特点为：

- 1) 主体为层次模型，整体关系为图结构，强关系构成树结构：
 - 层次模型：网格中存在多级行政管理部门和管理部门所管理的企业，各部门由于行政级别不同形成一个规整的层次模型；
 - 弱关系的图结构：临级部门之间，部门与企业之间，存在纵向关系，企业之间，同区县不同部门之间，不同区县同类部门之间存在横向关系；
 - 强关系树结构：行政管理部门间，企业与分管部门间都存在严格的行政管理机制，关系强于上面一点横向关系，故称为强关系，强关系形成树结构；
- 2) 存在单一的核心节点（市管理局）；
- 3) 同层中同类部门之间、同类部门与市管理局主管该类部门存在关系，并存在资源共享与资源传输；

- 4) 底层节点为乡镇企业，是郊区工业网的服务实体，既作为使用者接入网格，使用网格中的信息与数据，同时又作为资源提供者，为网格提供信息资源与数据资源；
- 5) 乡镇企业间存在横向与纵向的资源交互、共享与传输。例如，同行业企业会存在资源的共享与协同，处于生产链中的企业会以纵向的方式进行数据交互。

3、网格作业特点：

郊区工业网作为连同管理部门与乡镇企业，以及为乡镇企业提供服务的平台，其网格作业可分为三大类

- 1) 行政服务类：为用户提供行政类服务，简化用户各种项目申请审批、资格审核等行政类工作的流程，郊区工业网将各种行政类工作分类，并根据 workflow 对工作进行细化，提高作业的并行处理能力。例如，技改贴息项目中每个申请作业可能需要区县管理部门（3 个），区县总管（1 个），市管理部门（2 个），市管理局（1 个）多级多个部门的报表、材料，并提交回各部门做审批处理。用户提出请求后，郊区工业网根据作业要求从各个部门的节点中提取相关申请材料与说明返回给用户，用户填写完成提交后，系统将各资料发送到相关部门进行审核。出现问题根据 workflow 暂停相关作业，返回用户修改，修改后继续作业的处理。
- 2) 数据服务类：各企业拥有数据发布接口，可将本企业信息共享于郊区工业网中；各级管理部门又对其分管企业进行运营数据采集，完成定期统计工作，然后将可公布数据发布于郊区工业网中，为企业的运营提供了数据服务。企业间数据的共享，企业与管理部门数据的互通形成了郊区工业网的数据服务；
- 3) 计算服务类：利用郊区工业网中的各类服务器与计算设备为企业提供网格计算服务，以提高企业产品开发效率，缩短产品开发周期。

第二节 GOIDS 在天津市郊区工业信息网的应用

4.2.1 GOIDS 在天津市郊区工业信息网的应用

根据郊区工业网的网格节点与结构特点，GOIDS 运行在基本检测模式下，

在郊区工业网中进行如下配置：

1. 模块配置：

- 所有节点配置信息采集模块；
- 市管理局为郊区工业网网格环境的核心节点，配置 GOIDA 对整个 GOIDS 模块进行管理，并配置信息分析模块；
- 固定节点包括市乡镇管理局，各区县管理局，配置安全警戒模块、报警通道、信息共享模块与信息分析模块；
- 动态性中等节点包括市管理局部分部门、区县管理局部分部门，配置安全警戒模块、报警通道；
- 动态性强节点为企业节点，不要求配置除信息采集模块外的其它模块，但可根据自身处理能力选择性配置安全警戒模块与报警通道。

2. 结构配置：

市管理局服务器为网格环境的核心节点，配置 GOIDA，记录了网格中所有节点信息及网格中所有 GOIDS 模块配置情况，根据郊区工业网的特点，GOIDA 中对 $w = \alpha N + \beta L + \gamma C$ 进行相关设置

- 调用网络测速接口得到各个包含信息采集模块的节点到安全警戒模块所在节点，以及所有信息共享模块间与包含信息分析模块所在节点间的网速 N ，其他节点间网速为 ∞ ；
- 郊区工业网结构特点中强关系 $L=1$ ，弱关系 $L=2$ ，不存在明显关系 $L=4$ ；
- 初始 C 设置为 0，系统运行后 $C = (\text{GOIDS 模块所占内存} / \text{节点剩余内存}) \times 10$ ，与 N 、 L 数据级相匹配；
- $\alpha = 0.5$ ， $\beta = 0.3$ ， $\gamma = 0.2$ 。

4.2.2 系统运行及测试结果

1. GOIDA 负载均衡测试

1) 测试环境与方法

由于郊区工业网服务器分布于天津市各个区县，进行负载能力测试难于收集数据，通过网络集中收集很难保证数据具有时效性，且增加了 GOIDS 的开发成本，所以本测试在实验室环境中进行，为接近郊区工业网实际运行效果，GOIDA 中各参数采用与郊区工业网中实际参数相同设置的方式。所

有负载均衡能力待测节点均选用 Windows 操作系统并配置安全警戒模块，这样便于数据采集以及后期数据比较的公正性。

本测试选用了三个安全警戒节点，系统环境如下表所示：

表 4.1 测试节点环境列表

测试机编号	操作系统	CPU	内存
A	Windows Server 2003	P4 2.66	1GB
B	Windows Server 2003	P4 2.0	768MB
C	Windows 2000	P4 1.4	512MB

选用 12 台计算机，其中配置信息采集模块，编号为 1-12。测试采用轮转算法（1 配置 A，2 配置 B，3 配置 C，4 配置 A……，以此类推）、随机算法、GOIDA 三种方式对 12 个信息采集模块与待测安全警戒节点进行关联，并记录下待测节点的负载系数；同时信息采集模块采用逐个加入测试环境的方式进行，以检测系统的扩展能力。GOIDS 采用轻量级入侵检测软件为核心，CPU 占有时间较短很难进行比较，所以负载系数采用 GOIDS 所有模块内存占有量与系统剩余内存进行比较，待测机结束其他进程只运行 GOIDS。

2) 测试结果

对 12 个信息模块节点分别按照轮转算法、随机算法、GOIDA 三种算法与三台待测安全警戒节点机进行了测试，测试结果如表 4.2 中所示，其中横轴为加入信息检测节点的个数，纵轴为不同算法在不同数量节点时 A、B、C 三台机器的负载情况，“总”表示了三个待测节点的总负载情况，以衡量此时整个系统的运行状态。其中表格中 (X,Y) 表示此时该节点关联 X 个信息采集模块，负载系数为 Y，如 (3,1.38) 表示有 3 个信息采集模块关联到该节点，节点此时负载系数为 1.38。

表 4.2 测试结果

		1	2	3	4	5	6
轮转	A	(1,0.2)	(1, 0.2)	(1, 0.2)	(2, 0.34)	(2,0.34)	(2,0.34)
	B	(0,0)	(1,0.29)	(1, 0.29)	(1, 0.29)	(2,0.50)	(2,0.50)
	C	(0,0)	(0,0)	(1,0.56)	(1, 0.56)	(1, 0.56)	(2,0.98)
	总	0.2	0.49	1.05	1.19	1.40	1.82
随机	A	(1, 0.2)	(1, 0.2)	(1, 0.2)	(1, 0.2)	(1,0.2)	(2,0.34)
	B	(0,0)	(0,0)	(0,0)	(1, 0.29)	(2, 0.50)	(2, 0.50)
	C	(0,0)	(1, 0.56)	(2, 0.98)	(2, 0.98)	(2,0.98)	(2,0.98)
	总	0.2	0.76	1.18	1.47	1.68	1.82
GOIDA	A	(1, 0.2)	(2, 0.34)	(3, 0.46)	(5, 0.57)	(4, 0.57)	(4, 0.57)
	B	(0,0)	(0,0)	(0,0)	(0, 0)	(1, 0.29)	(2, 0.50)
	C	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	总	0.2	0.34	0.46	0.57	0.86	1.07
		7	8	9	10	11	12
轮转	A	(3,0.46)	(3, 0.46)	(3, 0.46)	(4, 0.57)	(4, 0.57)	(4, 0.57)
	B	(2, 0.50)	(3, 0.70)	(3, 0.70)	(3, 0.70)	(4,0.86)	(4, 0.86)
	C	(2, 0.98)	(2, 0.98)	(3,1.38)	(3, 1.38)	(3, 1.38)	(4, 1.71)
	总	1.94	2.14	2.54	2.65	2.81	3.14
随机	A	(2, 0.34)	(3, 0.46)	(4, 0.57)	(4, 0.57)	(4, 0.57)	(4, 0.57)
	B	(2, 0.50)	(2, 0.50)	(2, 0.50)	(2, 0.50)	(3,0.70)	(3, 0.70)
	C	(3, 1.38)	(3, 1.38)	(3, 1.38)	(4,1.71)	(4, 1.71)	(5,2.01)
	总	2.22	2.34	2.45	2.78	2.98	3.28
GOIDA	A	(4, 0.57)	(5,0.67)	(5, 0.67)	(5, 0.67)	(6,0.78)	(6,0.78)
	B	(2, 0.50)	(2, 0.50)	(3, 0.70)	(3, 0.70)	(3, 0.70)	(4, 0.86)
	C	(1, 0.56)	(1, 0.56)	(1, 0.56)	(2, 0.98)	(2, 0.98)	(2, 0.98)
	总	1.63	1.73	1.93	2.35	2.46	2.62

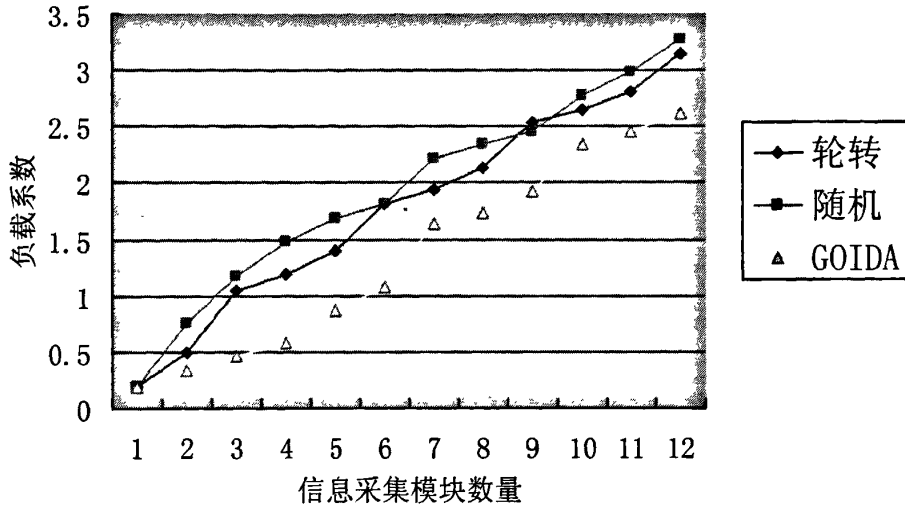


图 4.1 系统负载比较

信息采集模块以逐个加入测试环境的方式进行测试，均得到了测试结果，说明 GOIDS 具有较好的扩展能力，可在模块加入网格环境时自行扩展系统。

通过对三种算法系统总负载情况进行描点绘图（图 4.1）可以看出：轮转算法与随机算法负载总量相当，由于测试节点性能的差异导致节点间的负载均衡程度不高；而采用 GOIDA 使这个系统一直保持了一个较低的运行开销，并且通过表 4.2 中的数据可以发现采用 GOIDA 时系统中各节点保持了一个相对平衡的负载情况。

通过上述实验可以说明，通过 GOIDA 对 GOIDS 进行配置可以使系统具有良好的负载均衡能力，可以有效降低整个系统的总运行负担，以提高整个 GOIDS 的运行效率。

第五章 总结与展望

近年来,网络技术进入了高速发展时期,目前只通过安全通信与安全认证为网络提供安全保障是远远不够的。入侵检测系统作为网络中重要的安全保障手段,为网络提供主动防御的安全机制。网络环境与入侵检测系统相结合是网络安全的发展趋势,已经开始受到世界上一些学者的关注,出现了一些模型,但这些模型普遍存在着扩展性不足、自适应能力差、没考虑负载均衡等问题。

本文针对现有网络中入侵检测系统研究的不足,提出了一种面向网络的入侵检测系统 GOIDS 及该系统的负载均衡算法,并对 GOIDS 与负载均衡算法进行了设计、实现、实际部署与测试。本文所做的主要工作有:

- 1) 对入侵检测技术与现有网络安全技术进行了研究,论述了这两类技术分别在解决网络安全问题中的不足,提出了研究面向网络环境的入侵检测系统的必要性。
- 2) 通过分析了网络中常见的安全问题与入侵方式,总结了网络中入侵检测系统的设计需要。
- 3) 提出面向网络的入侵检测系统 GOIDS,对该系统整体构架及系统中各模块进行了详细的分析与设计。
- 4) 针对 GOIDS 并结合网络环境特点提出负载均衡算法 GOIDA,算法包括网络环境的模型化与基于图论理论的负载均衡算法两部分,并提出一个 GOIDA 的解决方案,结合解决方案对算法进行了说明。
- 5) 对 GOIDS 进行了实现,介绍了实现系统主要应用的相关技术,并重点介绍了 GOIDS 中接口的实现方式。
- 6) 根据天津市郊区工业信息网的网络环境特点对 GOIDS 进行了相应的设定,并运行于该网络中,对该网络提供了入侵检测服务。通过实际使用的结果与相关实验验证了 GOIDS 能适应网络环境,进行入侵检测,并具有良好的扩展性与负载均衡能力。

本文对网络环境中的入侵检测系统进行了深入的研究,提出了一个面向网络的入侵检测系统与相关的负载均衡算法,该系统具有具有扩展性强、动态负载均衡、拥有自适应能力的特点,该系统较好的解决了入侵检测系统难于适应网

格环境的问题，并在一定程度上弥补了现有网络安全技术的不足。但由于本人时间与工作能力有限，本文的研究还存在不足与待改进的地方，主要包括：

- 1) 现有的研究主要针对了面向网格入侵检测系统的扩展能力与负载均衡能力上，在自适应能力上只是采用了现有的传统分布式的基于行为入侵检测系统模型以保证 GOIDS 系统的完整性，未对 GOIDS 提供自适应方面的服务；
- 2) GOIDA 中各参数使用初始设置，并未对系统运行情况的返回数据进行分析，缺少对参数进行再优化的工作。

今后的主要工作将针对上面提到的不足开展：

- 1) 以网格中入侵检测系统自适应能力为重点，围绕基于网格行为的入侵检测模型进行研究并实现于系统中；
- 2) 以郊区工业网为基础对 GOIDA 各参数进行调试，对系统进行优化。最终实现在扩展性、负载均衡能力、自适应能力三方面都适合于网格环境的入侵检测系统。

参考文献

- [1] Foster I, Kesselmae. 网格计算(第二版). 金海等译. 北京: 电子工业出版社, 2004
- [2] Foster I, Berry D, Djaout A. The open Grid Services Architecture Version 1.5. <http://forge.gridforum.org/projects/ogsa-wg>, 2006.
- [3] 都志辉, 陈渝, 刘鹏. 网格计算. 北京: 清华大学出版社, 2002
- [4] 胡昌振主编. 网络入侵检测原理与技术. 北京: 北京理工大学出版社, 2006
- [5] 石志国, 贺也平, 赵悦. 信息安全概论. 北京: 清华大学出版社, 2007
- [6] Maozhen Li, Mark Baker. 网格计算核心技术. 王相林, 张善卿, 王景丽译. 北京: 清华大学出版社, 2006
- [7] 殷锋. 网格关键技术及校园网格应用研究. 四川: 西南交通大学出版社, 2007
- [8] Choon O, Samsudin A. Grid-based intrusion detection system. Proceedings of the 9th Asia-Pacific Conference on Communication. Malaysia, 2003. 1028~1032
- [9] Tolba M, Abdel Wahab M, Taha I A, et al. GIAD: Toward enabling grid intrusion detection systems. Proceedings of the 5th IEEE/ACM Int. Symp. on Cluster Computing and the Grid(CCGrid2005). Cardiff, UK, 2005.
- [10] Kenny S, Coghlan B. Towards a grid-wide intrusion detection system. Proceedings of the European Grid Conferenee(EGC2005). Amsterdam, Netherlands, 2005. 275~284
- [11] Fangyie L, JiaChun L, MingChang L. A Performance-Based Grid Intursion Detection System. Proceedings of the 29th Annual IEEE International Computer Software and Applications Conference. Edinburgh, 2005. 525~530
- [12] Fang-Yie L, Ming-Chang L, Jia-Chun L, Chao-Tung Y. Detection workload in a dynamic grid-based intrusion detection environment. Journal of Parallel and Distributed Computing 2008, 04: 427~442
- [13] Sliva P, Westphall C B, Westphall C M, Assuncao M. Composition of a

- DIDS by Integrating Heterogeneous IDSs on Grids. 4th international workshop on middleware for grid computing (MGC'06). Melbourne, Australia, 2006. 109~115
- [14] Schulter A, Reis J A, Koch F, Westphall C B. A Grid-based Intrusion Detection System. Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006(ICN/ICONS/MCL 2006). Mauritius, 2006.187~187
- [15] Yu-Xin W, Mu-Qing W. An Intelligent Grid Intrusion Detection System. Journal of Software, 2006, 11: 2385~2394
- [16] 张雪松. 一种新的高速网格入侵检测系统模型. 科学技术与工程, 2008, 01: 91~95
- [17] 段丽艳, 刘波, 余小华. 面向网格的分布式入侵检测模型研究. 现代计算机 (专业版), 2007, 08: 44~46
- [18] 陈树伟, 刘传领. 一种基于 XML 的入侵检测消息的处理机制. 商丘师范学院学报, 2007, 09: 86~88
- [19] Bangyi L, Enyu Y. Complexity and Algorithm of the Minimum Spanning Tree Problem with Multiple Parameter. CONTROL AND DECISION, 2000, 05: 617~619
- [20] 林大志. 具有任意多个引出点的约束最小支撑树问题. 河南科学, 2007, 05: 722~724
- [21] Snort. <http://www.snort.org/>
- [22] XML Schemas. <http://xml.coverpages.org/schemas.html>
- [23] World Wide Web Consortium (W3C). <http://www.w3.org/>
- [24] 顾兵主编. XML 实用技术教程. 北京: 清华大学出版社, 2007

致 谢

首先感谢我的导师邵秀丽教授。在我三年的研究生学习期间，邵老师严谨的治学态度、积极努力的工作精神为我们树立了光辉的榜样。每一次您对我的教诲，都是我人生路上的一个启迪，这是课本中无法学到的，将使我受益终身。论文期间，从论文选题、模型的建立、关键点突破，到论文撰写与最终定稿工作，每一个阶段邵老师都给予了我极大的帮助与指导，每一次和邵老师的讨论都让我受益匪浅。在这里，我衷心的祝您身体健康，家庭幸福！

感谢我的父母，是您们给予我生命、抚育我成人，给予我无私的爱，为我的成长付出了无数心血。希望您们身体健康，永远快乐。

感谢我的女朋友盛鸣，研究生在读的三年期间默默守护等待着我，并且在论文期间给予了我极大的支持。感谢你对我一如既往的关怀。

感谢实验室的兄弟姐妹，从他们那里我学到了许多。希望在这几年的共同学习中我也给你们带来了快乐。祝你们学业有成。

最后感谢我的室友王昊明、付翔、袁振坤，共同的幽默建立了宿舍轻松快乐的氛围，这使得我的生活学习更加丰富多彩。毕业在即，但此段经历终生难忘。祝你们工作顺利。

个人简历

一、个人基本情况

姓名 孟晋津
性别 男
出生日期 1982年3月25日
所在学校 南开大学
所学专业 计算机软件与理论

二、获得学位情况

时间	学位	毕业院校	所学专业
2004.7	学士	南开大学信息学院	计算机科学与技术

三、研究生在读期间主要参与的工作

2005.7 - 2006.1	南开之星的 Web 作业提交和管理系统
2005.10 - 2006.5	生物医药高性能并行计算及创新应用平台
2006.5 - 2006.12	南开之星公共服务平台
2006.11 - 2007.9	企业信息化应用平台
2007.4 - 现今	天津市郊区工业信息网