 密级：

硕 士 学 位 论 文



论文题目

作者姓名

指导教师 教授

教授

学科(专业)

所在学院

提交日期

A Dissertation Submitted to Zhejiang University for the Degree of

Master of Engineering



TITLE:

Author:

Supervisor:

Subject:

College:

Submitted Date:

摘要

**关键词**

**Abstract**

**Keywords**

目录

[摘要 i](#_Toc165262352)

[**Abstract** ii](#_Toc165262353)

[第1章 绪论 2](#_Toc165262354)

[1.1 课题背景 2](#_Toc165262355)

[1.1.1 2](#_Toc165262356)

[1.2 本章小结 2](#_Toc165262357)

[1.2.1 2](#_Toc165262358)

[第2章 2](#_Toc165262359)

[2.1 第一节 2](#_Toc165262360)

[2.1.1 2](#_Toc165262361)

[2.2 本章小结 2](#_Toc165262362)

[2.2.1 2](#_Toc165262363)

[第3章 2](#_Toc165262364)

[3.1 第一节 2](#_Toc165262365)

[3.1.1 2](#_Toc165262366)

[3.2 本章小结 2](#_Toc165262367)

[3.2.1 2](#_Toc165262368)

[第4章 2](#_Toc165262369)

[4.1 第一节 2](#_Toc165262370)

[4.1.1 2](#_Toc165262371)

[4.2 本章小结 2](#_Toc165262372)

[4.2.1 2](#_Toc165262373)

[第5章 2](#_Toc165262374)

[5.1 第一节 2](#_Toc165262375)

[5.1.1 2](#_Toc165262376)

[5.2 本章小结 2](#_Toc165262377)

[5.2.1 2](#_Toc165262378)

[第6章 2](#_Toc165262379)

[6.1 第一节 2](#_Toc165262380)

[6.1.1 2](#_Toc165262381)

[6.2 本章小结 2](#_Toc165262382)

[6.2.1 2](#_Toc165262383)

[第7章 2](#_Toc165262384)

[7.1 第一节 2](#_Toc165262385)

[7.1.1 2](#_Toc165262386)

[7.2 本章小结 2](#_Toc165262387)

[7.2.1 2](#_Toc165262388)

[第8章 2](#_Toc165262389)

[8.1 第一节 2](#_Toc165262390)

[8.1.1 2](#_Toc165262391)

[8.2 本章小结 2](#_Toc165262392)

[8.2.1 2](#_Toc165262393)

[参考文献 2](#_Toc165262394)

[攻读硕士学位期间主要的研究成果 2](#_Toc165262395)

[致谢 2](#_Toc165262396)

图目录

[图 1‑1简单的语音信号多描述编解码过程 2](#_Toc164669029)

[图 3‑1 流程图 2](#_Toc164669030)

表目录

[表 2‑1简单的多描述分配表 2](#_Toc164669160)

# 绪论

## 课题背景

基于僵尸网络的分布式拒绝服务（DDos）攻击通过泛洪选定的因特网服务器已经被使用过很长时间。然而，可以有效断开因特网服务器的链路泛洪攻击仍然非常罕见。与传统的DDos攻击终端服务器不同的是，这类攻击可以导致路由的不稳定以及互联网连接的中断。特别是攻击目标是能源分配，实时性的金融，命令和控制服务等关键设施时，链路泛洪攻击会非常有效。过去的分布式拒绝服务攻击（DDos）的共同点是直接对受害目标发起攻击，因此，很多针对此类攻击的防御机制会采取识别大量流量的来源或者优先考虑合法流量。针对伪造源地址的流量，很多追踪方案已经开发出了识别恶意流量的起始网络使得ISP一旦发现后可以直接处理。但是在使用大型的僵尸网络后，攻击者依靠DDos攻击生成少量的恶意流量，导致受害者难以区分合法流量和恶意流量。为了处理这种隐秘的攻击，基于容量的系统允许终端主机识别长期运行的合法流量，这些流量会被路由器优先传送。这样，在高负载期间，路由器转发合适容量的数据包，同时丢弃不合适的数据包。部署了追踪和流量容量功能后，攻击者为了找到新的方式来启动DDos攻击，把攻击目标设置为网络中关键链路而不是终端主机。在这样的背景下，链路泛洪攻击便由此产生，根据现有的工作可知，禁用重要的链路会在隔离网络的各个部分方面造成重大损失。本文主要研究新型DDoS攻击的作用，这种攻击可以逃过传统的DDoS防御并关闭关键链路。为了绕过试图消除恶意流量的传统DDoS防御系统，链路泛洪攻击中的僵尸网络仅发送有用流量或合法流量，这些合法流量会耗尽核心网络链路的网络带宽。结果，需要通过这些拥塞的关键网络链路的合法主机的流量将受到严重影响。

LFA是很难被检测的，因为目标链路是由攻击者选择的，可能位于与包含目标区域的自治系统不同的自治系统中，并且攻击流量不会到达目标区域，因此受害者很难检测到攻击。其次，每个僵尸主机通过发送低速率的合法流量到公共服务器，因此普通的DDos检测系统无法识别。僵尸程序可以更改其流量模式，从而规避基于异常流量模式的系统。 尽管已经提出了一些基于路由器的方法来防御此类攻击[Towards Detecting Target Link Flooding Attack-6-8]，但由于无法立即将其广泛部署到Internet上，因此其有效性可能受到限制。 在这样的背景下，如何针对LFA进行检测是当前学术领域关注的焦点。

## 国内外研究现状

链路泛洪攻击（LFA）作为一种新的DDos攻击类型，在该领域中已经有了不少研究工作。本小节主要总结链路泛洪攻击模型的研究和链路泛洪攻击检测的相关工作。

### LFA模型及特点

LFA主要被分为coremelt攻击和crossfire攻击两种。在coremelt攻击中，僵尸主机互相发送数据包来阻塞一组AS的骨干路由器，主要特点是只会创建所需的流量，因此无法规避所有防御过滤有害流量的机制，并且需要调节大量的僵尸主机对来让他们的通信路径共享网络的关键链路，但是可以不受TCP的拥塞控制机制的约束发送更大的流量。而Crossfire的攻击终端是诱饵服务器而不是僵尸主机，必须发送不受拥塞控制的合法协议消息，比如http的get请求，并且可以仅发送低密度的流量到任意公共ip地址的主机上，这些主机被当做诱饵服务器，从而泛洪任何选定的关键链路而不用受限于僵尸主机的分布，并且攻击流无法被区分。当前部署的大多数防御机制都可以保护单个企业服务器或其与Internet的网络链路免受服务器泛洪的攻击。但是，当针对企业的攻击通过泛洪选定的上游网络链路以降低服务器与Internet服务器的连接时，传统的企业防御机制（例如防火墙或入侵检测/保护系统）将无法再保护服务器。这些攻击只能在企业上游得到缓解。即ISP或内容分发网络（CDN）。但是，到目前为止，这些防御措施都不能用来抵抗大规模的LFA。

### LFA检测的研究现状

国内外关于LFA检测和缓解已经进行了许多工作，主要分为基于流量过滤机制，基于BGP等网络协议以及路由决策，最后是结合SDN控制器来进行LFA检测。

#### 基于流量过滤机制

先前针对链路泛洪攻击的大多数对策都是基于过滤的机制，该机制可识别攻击流或流聚合，并在目标路由器和上游路由器上安装过滤器以阻止多余的流量。通过[codef-16]的pushback方法，目标路由器可以识别导致其泛洪的流聚合，并要求上游路由器安装过滤器以阻止高速率流。此外，上游路由器对其他上游路由器递归执行推回操作，最终直接为攻击源提供服务的路由器会安装过滤器。 [13，19]中的最新提议提出了在发起攻击流量的AS处直接安装过滤器的新方法。尽管基于过滤的方法可能可以有效抵御传统攻击，但不能将其用于缓解使用低速率攻击流进行的大规模，链路泛洪的攻击。实际上，当用于减轻此类攻击时，基于过滤的方法始终会对合法流造成附带损害的高风险。在[deep learning cfa-8]中，作者提出了一种使用集中式流量层次的控制和监视系统来克服LFA的方法。在此模型中，防御者通过在链路和路径之间执行负载平衡来尝试保持网络正常运行而不会出现任何拥塞。防御者还记录导致流量拥塞的攻击源，并在超过累积流量阈值时对流量进行速率限制。在[10]中，作者使用类似的方法提出将网络划分为多个域并在本地域级别或域间级别执行流量重新路由。但是，这些工作没有考虑流量之间的时间相关性，如果交通流量也超过阈值，使用阈值方法可能会影响合法流量。

#### 基于网络协议和路由决策

[codef-4]提出了CoDef（协作防御），它是一种AS（自治系统）级目标链路洪泛攻击避免方案。相比前面的方法，Codef不会安装过滤器，从而避免了附带损害的根源，即使它能够非常准确地将攻击流与合法流区分开。当发生目标链路泛洪攻击时，拥塞的路由器会向源AS发送重新路由请求和速率控制请求，以缓解链路拥塞。这是因为攻击者必须将攻击流聚合到目标链路才能成功进行攻击。而且，拥塞的路由器通过监视源AS的行为来区分合法的AS与攻击的AS。结果，该方案迫使攻击者做出不合理的选择：要么遵循请求并放弃目标链路上的攻击持久性，要么拒绝请求并遵循流控制，并且最大的缺点是泛用性不高不能实现快速部署。

还有一种方法是多种多径路由机制，以通过多路径转发流量来减少或避免拥塞[codef-6,9,25,33,36]。这些机制尽管在实现细节上不同（例如，覆盖路由，域间路由或域内路由），但是它们通过使各个路由器根据本地拥塞信息来做出路由决策而依赖于路径分集来分散流量。然而，仅出于两个原因，仅路径分集根本不足以处理大规模洪水攻击。第一个是攻击防御扩展的不对称性：与多路径配置相比，由分布式僵尸机发起的攻击可以更快，更便宜地进行扩展。因此，在缺少诸如流可区分性的附加机制的情况下，多路径路由不能提供足够的防御，其中流区分性将合法流和攻击流分开，然后以不同方式对其进行处理（例如，路由）。第二个原因是多路径防御，它仅在不区分流量的情况下转移不同路径上的流量，只会分散攻击流。因此，攻击流将继续影响合法流（例如，减少可用带宽）并在下游聚合，从而使识别和处理更加困难。这种方法的有效性可能会受到限制，因为它们无法立即广泛部署到Internet。

在[deep learning cfa-9]中，作者开发了LinkScope，该系统用于检测链路泛洪攻击并定位目标区域或链路，LinkScope不会被动检查流量以发现异常，而是进行非合作的主动测量以覆盖尽可能多的路径，并捕获LFA对性能指标的负面影响。这项工作是检测由LFA引起的严重拥塞的极好方法，但是它需要部署许多附加的探测点。 此外，这项工作没有考虑所有拥塞链路的分布，并且缺乏减轻或防御LFA的相关对策。文献[deep-9]中提出的系统使用逐跳和端到端网络测量技术，这些技术通过测试数据包探测网络并测量流量特性以检测性能下降。这将向网络添加冗余流量。

在[BGP]中，作者提出了Nyx系统，通过利用自治系统（ASes）处理现有边界网关协议（BGP）中的路由通知的方式，Nyx允许部署中的AS隔离来自关键上游AS的流量，使其脱离受攻击的链路，并进入备选的，未拥塞的路径。这种隔离消除了对攻击流量进行过滤或取消优先级的需求。 Nyx通过正常的BGP路径选择来控制出站路径，而来自关键AS的返回路径则通过使用本文使用现有流量工程原理开发的特定技术进行控制，而无需外部协调。作者将备用路由传播到部署者的网络并控制哪些AS通过策略触发的BGP环路检测来传播这些路由，通过有选择地掩盖路径上出现的网络实现了超出部署者控制范围的路由器中的环路检测。Nyx使用现有的流量工程技术使替代路径在数据包转发方面更为可取，并利用FRRP确保替代路径在DDoS攻击下绕过链路传播，但从未真正到达这些链路。最后，Nyx系统确保生成的替代路径具有足够的备用容量，以处理来自关键网络的流量以及来自任何受干扰网络的流量。Nyx通过撤消替代路径并尝试对其进行重新传播来对新的潜在替代路由进行采样，从而避免将路由传播到DDoS攻击下的两个链路以及先前替代路径中的拥塞链路。这种方法将针对LFA的流量过滤问题转成路由问题，避开了检测直接通过BGP路由缓解LFA但是该方法只能针对当个AS进行保护缺乏扩展性。

#### 基于SDN控制器

在[wood]中，作者提出了Woodpecker方案，该方案可以通过SDN有效缓解LFA。首先，作者提出了一种增量部署方案，该方案可以最大化网络的可用路径。然后，使用预定义的规则作为触发来检测拥塞并在控制器和升级的设备的帮助下探测拥塞的链接。设计了一种识别算法，根据全局拥塞信息来判断LFA是否引起了拥塞。最后，为减轻LFA的影响，Woodpecker采用了基于升级节点的集中式流量工程，从而使流量足够均衡，并消除了路由瓶颈。另外，作者设计了算法：当LFA超出网络容量时，将根据黑名单丢弃某些数据包作为紧急措施。此方案使用启发式算法来选择一组交换机，以升级到1040个启用SDN的交换机。在全局视图和数据平面触发器的帮助下，Woodpecker可以快速定位拥塞并通过全局拥塞信息确定LFA是否导致拥塞。为了减轻LFA，Woodpecker执行全局流量工程，以消除阻塞链接。

在【lfa-SDN】中，作者提出了LFADefender利用SDN提供的关键功能来检测和缓解LFA，使用SDN技术，LFADefender可以轻松识别LFA攻击者可能使用的目标链路，因为SDN控制器具有网络拓扑的全局视图，并且可以动态跟踪网络中的流路，本文借鉴了这一点使用了SDN控制器结合链路选择算法找到关键链路，作者通过使用高性能采样技术sFlow [9]，在LFADefender中设计了一个链路拥塞监控器。可以将链路拥塞监控器动态部署在任何目标链路的每一端，以有效地捕获流量数据，然后将其发送到SDN控制器，在该SDN控制器中对其进行分析，以生成丰富，实时的全网络流量视图。该方法通过SDN监测到链路阻塞来检测LFA，实时性较差。

## 本文研究内容

本文的主要研究内容是仿真网络模拟LFA并证明LFA的有效性，然后提出了基于SDN网络架构的LFA的检测方法，利用SDN和深度学习算法针对LFA进行有效地攻击检测。

本文的主要研究内容如下：

1.研究了基于SDN网络架构的LFA实现

本文通过OVS模拟交换机和namespace模拟主机借助多台服务器搭建网络拓扑，路由算法基于最短路径，基于泊松分布生成背景流量，带宽模型采用固定带宽通过tc工具来控制网卡的发包速率，基于贪心算法来选择关键链路，在此基础上通过选定僵尸主机和诱饵服务器来完成LFA，在僵尸主机上用hping3工具来生成攻击流量。通过Opendaylight来作为SDN控制器，利用SDN控制器可以为LFA检测选择关键链路以及采集数据，网络拓扑的搭建与直接利用mininet仿真不同更加真实地模拟了现实中的网络环境，测试使用iperf和ping工具通过丢包率，RTT和带宽三个指标验证了LFA的有效性。

2.研究了基于SDN网络架构的LFA检测

## 本文组织架构

本文章节主要分为七章，内容分别如下：

第1章 绪论，

第2章 相关技术及原理

第3章 基于SDN的链路泛洪攻击模拟实现

第4章 基于SDN的链路泛洪攻击检测方法

第5章 基于SDN的链路泛洪攻击与检测系统设计

第6章 实验测试与评估

第7章 总结与展望，总结了本文主要研究的工作成果和实验中的问题，并对需要解决和优化的要点明确下一步的工作研究方向。

# 相关技术及原理

本章主要围绕基于SDN网络架构的LFA实现和检测这个研究方向，介绍了跟本文密切相关的几个领域的背景知识和技术原理。本章将着重介绍SDN技术、LFA原理以及深度学习技术三个方面。

## SDN技术

### SDN架构

软件定义网络（Software-Defined Network，SDN）经斯坦福大学Nick McKeown教授的研究团队提出，并且由ONF提供了最明确和广为接受的SDN定义:软件定义网络（SDN）是一种新兴的网络体系结构，其中网络控制与转发分离，并且可以直接编程。根据此定义，SDN由两个特征定义，即控制平面和数据平面的解耦以及控制平面上的可编程性。[a survey on sdn ]SDN的架构由基础设施层、控制层和应用层等三层组成，如图2.1 所示。

基础设施层位于底部，也可以叫做数据平面。基础设施层由数据平面中的交换设备（例如，交换机，路由器等）组成。这些交换设备的功能大部分是双重的。首先，他们负责收集包括诸如网络拓扑，流量统计和网络使用等网络状态，将它们临时存储在本地设备中，然后将其发送给控制器，根据控制器提供的规则处理数据包。

中间层是控制层，也可以叫做控制平面。控制层通过其两个接口桥接应用程序层和基础结构层。为了与基础设施层（即，南向接口）进行向下交互，它指定了由交换设备提供的给控制器访问的功能。这些功能包括报告网络状态和导入分组转发规则。为了与应用程序层（即北向接口）进行向上交互，它提供了各种形式的服务访问点，例如应用程序编程接口（API）。SDN应用程序可以通过此API访问交换设备报告的网络状态信息，根据此信息做出系统调整决策，并通过使用此API为交换设备设置数据包转发规则来执行这些决策。由于一个大型管理网络域中将存在多个控制器，因此控制器之间还需要一个“东西方”通信接口，以便控制器共享网络信息并协调其决策过程。

最顶层是应用层，这是由SDN 网络的控制者定制的层面。应用程序层包含旨在满足用户要求的SDN应用程序。通过控制层提供的可编程平台，SDN应用程序可以访问和控制基础结构层的交换设备。SDN应用程序可以包括动态访问控制，网络安全，云计算服务，服务器负载平衡以及网络虚拟化。

### Openflow协议

OpenFlow由McKeown等人首先提出。目的是使校园网络中的网络实验变得容易，并且目前在大多数SDN实践中都使用。使用OpenFlow的早期实验主要旨在创建一个专注于控制数据包转发的独立软件可控制网络。后来，研究发现，使用OpenFlow实施单独的软件可控网络实际上是所谓的“软件定义网络”的实用促成因素。

OpenFlow充分利用了以下事实：大多数现代以太网交换机和路由器都包含用于基本联网功能的流表，例如路由，子网划分，防火墙保护以及数据流的统计分析。 在OpenFlow交换机中，流表中的每个条目都包含三个部分，“头”用于匹配接收到的数据包，“动作”用于定义对匹配的数据包执行的操作，以及匹配流量的“统计”。OpenFlow协议为控制器提供了方便的流表操纵服务，以便控制器通过安全的TCP通道远程插入，删除，修改和查找流表条目。一个OpenFlow交换机包括一个或多个流表和一个组表，来执行数据包查找和转发，以及一个到外部控制器的OpenFlow通道（图1）。交换机与控制器通信，并且控制器通过OpenFlow协议管理交换机。使用OpenFlow协议，控制器可以以响应方式（响应数据包）和主动方式添加，更新和删除流表中的流条目。交换机中的每个流表都包含一组流条目。每个流条目均包含匹配字段，计数器和一组适用于匹配的指令数据包（请参阅5.2）。匹配从第一个流表开始，并且可以继续到其他流表（请参阅5.1）。流条目按优先级顺序匹配数据包，并使用每个表中的第一个匹配条目（请参阅5.3）。如果找到匹配的条目，则执行与特定流条目关联的指令。如果在流表中找不到匹配项，则结果取决于表未命中流条目的配置：例如，数据包可能会通过OpenFlow通道转发到控制器，可能会被丢弃，或者可能继续到下一个流表。

### SDN优点

在SDN中，网络智能逻辑上集中在基于软件的控制器（控制平面）中，并且网络设备成为可以通过开放接口（例如，ForCES [1]，OpenFlow）进行编程的简单数据包转发设备（数据平面）。

增强配置：在网络管理中，配置是最重要的功能之一。具体而言，当将新设备添加到现有网络中时，需要适当的配置以实现整体上一致的网络操作。但是，由于网络设备制造商和配置接口之间的异质性，当前的网络配置通常涉及一定级别的手动处理。此手动配置过程繁琐且容易出错。同时，还需要花费大量精力来对配置错误的网络进行故障排除。在传统网络的设计中，网络的自动和动态重新配置仍然是一个很大的挑战。 SDN将有助于纠正网络管理中的这种情况。在SDN中，控制平面在各种网络设备包括交换机，路由器，网络地址转换器（NAT），防火墙和负载平衡器上的统一，使得可以从单个点自动配置网络设备，软件控制。这样，可以基于网络状态以编程方式配置和动态优化整个网络。

改善性能：在网络运营中，主要目标之一是最大程度地利用已投资网络基础设施。然而，由于在单个网络中各种技术和利益相关者的共存，因此认为优化整个网络的性能很困难。当前的方法通常专注于优化网络子集的性能或某些网络服务的用户体验质量。显然，这些方法基于本地信息而无需跨层考虑，即使与网络操作不冲突，也可能导致性能欠佳。 SDN的推出为全球改善网络性能提供了机会。具体地说，SDN允许使用全局网络视图进行集中控制，并使用网络体系结构中不同层之间交换的信息进行反馈控制。这样，许多挑战性的性能优化问题将通过适当设计的集中式算法来解决。随之而来的是针对经典问题的新解决方案，例如数据流量调度，端到端拥塞控制，负载均衡数据包路由，节能运行和服务质量（QoS）可以开发并轻松部署支持，以验证其在改善网络性能方面的有效性。

促进创新：在网络应用程序不断发展的情况下，未来的网络应鼓励创新，而不是试图准确地预测并完全满足未来应用程序的要求。 不幸的是，任何新的想法或设计都将在实施，试验和部署到现有网络中立即面临挑战。 主要障碍来自常规网络组件中广泛使用的专有硬件，从而无法进行实验修改。 此外，即使有可能进行实验，也常常在单独的简化测试平台上进行实验。这些实验不能为这些新思想或网络设计的工业适应提供足够的信心。 诸如PlanetLab和GENI之类的社区努力背后的想法无法进行大规模实验，无法完全解决问题。相比之下，SDN通过提供可编程的网络平台来实施创新，进行实验并方便，灵活地部署新思想，新应用和新创收服务，从而鼓励创新。 SDN的高可配置性提供了虚拟网络之间的清晰分隔，允许在真实环境中进行实验。 可以通过从实验阶段到操作阶段的无缝过渡来逐步部署新想法。

## 链路泛洪攻击

### 链路泛洪攻击原理

传统的ddos攻击的主要方法有两种：一种常见方法是，攻击者将数据包流发送给受害者，这种流消耗了网络带宽等重要的资源，并使合法流量无法用于受害者；另一个常见的方法是，攻击者发送几个格式错误的数据包，这些数据包会混淆受害机器中的应用程序或协议，并迫使其冻结或重新启动。可以发现，传统的ddos攻击主要针对选择性地断开终端服务器，然而LFA会导致路由的不稳定和互联网连接中断。

LFA核心思想主要是通过泛洪精心选择的链路来隔离目标区域。通过选定僵尸机发送攻击流量到诱饵服务器，并且所有流汇聚到所选的目标链路上造成目标链路阻塞。由于攻击发生在链路并且诱饵服务器在目标区域的上游，所以在目标路由器或端点服务器上都没有有效的对策来检测和防御LFA。

与传统的DDoS攻击相比，LFA具有一些显着的功能。首先，被攻击的目标是不同的。在传统的DDoS攻击中，攻击者将大量意外流量直接发送到目标服务器。相反，LFA攻击者组织了许多僵尸程序，以将低速流量发送到诱饵服务器，并尝试淹没将目标区域连接到Internet的关键链路。因此，受害者甚至不知道它受到了攻击。其次，传统的DDoS攻击只会使一台服务器不可用，而LFA会阻止目标区域中的所有服务器访问Internet。第三，当前的DDoS防御系统试图消除不必要的流量，而LFA中的僵尸程序则发送想要的流量或合法流量。因此，传统的DDoS防御系统无法有效地区分攻击流量和正常流量。

能够成功发动LFA的主要原因是负责传送所有流量的大部分到特定区域的所选链路是很少的，这是由于互联网IGP/BGP协议通过最短路径进行路由的情况下，流量集中在距离目标区域几跳（例如三到四跳）的一小部分链路中，并且可以通过诸如traceroute之类的通用工具轻松发现。LFA是可以持久存在的，它无法通过单个ISP或通过终端服务器来阻止，因为ISP和服务器被至少有三跳远的泛洪链路所断开。在LFA中，攻击流量的构造应使仅在控制平面不受影响的情况下淹没链路的数据平面，因此只有在检测到数据平面泛洪之后才可以启动动态重新路由，这给了攻击者充足的时间选择同一目标区域的备用链路集；通过对到该链路的路由上的可用带宽进行事先估计，避免了位于目标链路上游的链路的早期拥塞；第三，分布在多个ISP上的多个不相交的目标链路集的可用性意味着没有一个ISP可以独立检测和处理LFA。

任何传统的DDos攻击检测和防御系统都无法轻易检测到LFA，主要原因有三点。首先，僵尸程序可以使用有效的IP地址，因此基于检测或防止使用欺骗性IP地址的防御就变得无关紧要；例如，基于入口过滤的防御[cfa-9]，功能系统[10、11]或负责的协议设计[12、13]。其次，僵尸网络可以在使用必要流量的情况下泛洪链路。例如，它们可以以针对路由器组的方式互相发送数据包[7]。第三，僵尸网络可以以低密度的流量发起攻击，这些流量会同时穿越目标链路并将其泛洪从而产生阻塞。例如，僵尸网络控制器可以计算出一大组IP地址（即诱饵IP），这些IP地址的对应路径跨越同一链路，然后指示其僵尸网络向这些地址发送低强度流量。LFA能够让位于诱饵IP地址的任何服务器无法检测到，并且ISP很晚才能看到其影响。此外，当前的流量工程技术无法抵抗这些攻击。离线流量工程的延迟非常高，而在线流量工程技术不能提供强大的稳定性保证[17]，尤其是当多个ISP需要协调其响应以应对攻击时，以及因此无法部署在Internet主干网中。更糟糕的是，即使可以部署在线技术，攻击者也可以实时更改目标链路的集合，从而绕过在线流量工程的防御。

### 链路泛洪攻击分类

常见的LFA有两种，一种是coremelt攻击，另一种是crossfire攻击，后者更加容易实施并且泛用性更强。Crossfire攻击是Coremelt攻击的升级，它引入了一些新概念：诱饵服务器，流量密度和滚动攻击。crossfire攻击使用僵尸机作为源，并使用一些公共服务器作为目标，通过源和目标之间的流量泛洪目标链路。 这种LFA在实际中更容易实施，因为它只需要不断地探测目标区域中的公共服务器来构建链路映射，而无需像Coremelt攻击对僵尸机的部署有严格的要求。Coremelt和Crossfire之间的主要区别在于后者不需要在目标区域中部署僵尸机来发送和接收流量以超载目标链路。相反，Crossfire中的僵尸机可以将合法流量发送到目标区域附近的诱饵服务器。这两类攻击的核心思想都是通过攻击目标链路来中断网络的连接从而间接地影响目标区域。

#### Coremelt攻击

Coremelt攻击中的僵尸网络仅发送有用流量或“合法”流量，通过成对的僵尸机之间的连接对发送。 因为在具有个僵尸主机的网络中存在个连接，所以这些“合法”流可能耗尽核心网络链路的网络带宽。 结果，需要穿越这些拥塞的核心网络链路的合法客户端的流量将受到严重影响。在Coremelt攻击中，攻击者使用一系列僵尸主机来相互发送数据，攻击的目标是几个网络域（自治系统-ASes）内的骨干链路。 在僵尸主机之间相互发送数据时，攻击者可以逃避基于容量和过滤的DoS防御系统，因为接收者需要收到所有的流量。 当僵尸主机分布在多个网络中时，攻击者更有可能关闭主干链路，而不会破坏较小的支路链路。 发起Coremelt攻击有3个步骤：在网络中选择一个核心链路作为目标链路；确定哪些僵尸主机对可以生成通过目标链路的流量；在步骤2中标识的僵尸主机对之间发送流量，以使目标链路过载。

#### Crossfire攻击

crossfire攻击包括目标区域，目标链路，诱饵服务器等概念。目标区域是攻击者对其发起攻击的Internet地理区域（引用cfa图），包括攻击者选择的的组织，城市，州，地区甚至一个国家。目标链路是攻击者需要泛洪的一组网络链路，以便将目标区域与Internet的其余部分隔开，这些精心选择的网络链路是泛洪攻击的实际目标，而目标区域是真实的预期目标。为了对目标区域发动Crossfire攻击，攻击者会选择目标区域内的一组公共服务器以及目标区域周围的一组诱骗服务器。这些服务器是从可公开访问的服务器中选择的，因此很容易找到它们。公共服务器组用于构建以目标区域为中心的攻击拓扑，诱饵服务器组用于创建攻击流。然后，攻击者构造一个“链路图”，即从其僵尸地址到公共服务器地址的第3层链路图。是一旦创建了链路图，攻击者就会使用它来选择最佳目标链路，其泛洪将有效地切断Internet的目标区域。接下来，攻击者设置僵尸主机到诱饵服务器的流量以淹没目标链路，这最终将阻止大多数发往目标区域的流量。发起Crossfire攻击所需的三个主要步骤包括链路图构建，攻击建立和僵尸主机的协调。

## 机器学习技术

机器学习是人工智能的一种应用，它使系统能够自动学习并从经验中进行改进，而无需进行明确的编程。在做DDos攻击检测方面，机器学习具有很大的优点，可以高效增加检测的准确度，减少错误率，同时有效降低计算成本。机器学习算法根据训练数据建立数学模型，以便进行预测或决策，在分类上通常分为有监督学习，无监督学习和半监督学习三种。

有监督学习算法，可以使用标记的示例将过去所学的知识应用于新数据以预测未来的事件。从对已知训练数据集的分析开始，学习算法会产生一个推断函数，以对输出值进行预测。监督学习问题可以进一步分为回归和分类问题。常见的有监督的机器学习算法是用于回归问题的线性回归以及解决分类和回归问题的随机森林算法和用于分类问题的支持向量机。由于支持向量机对于分类问题有极为有效且可行的解法，它被广泛应用于多种防御系统当中。其原理基于在高维空间中通过超平面划分解决分类问题，所以其核函数的选择至关重要。同时，支持向量机由于包含了大量计算，所以相对的资源消耗较大，比较占用计算单元和存储空间。随机森林算法相比于支持向量机，对于不均匀数据集有着更强的适应能力，通过将多颗决策树集成起来，通过对每棵子树的分类结果进行投票统计来获取最终结果。但随机森林在处理问题时，由于不能进行超出训练范围的预测，在某些噪音较大的回归或分类问题上，会出现过拟合的情况

无监督学习算法，主要研究系统如何从未标记的数据推断出描述隐藏结构的功能。当用于训练的信息既未分类也未标记时，将使用无监督的机器学习算法。系统无法找出正确的输出，但可以浏览数据并可以从数据集中得出推论，以描述未标记数据中的隐藏结构。无监督学习问题可以进一步分为聚类和关联问题。聚类：聚类问题是您要发现数据中固有的分组，例如通过购买行为对客户进行分组；关联：关联规则学习问题是想发现描述数据大部分的规则，例如购买X的人也倾向于购买Y。主要算法有聚类问题的k-means算法和关联规则学习问题的Apriori算法。

半监督学习算法介于有监督的学习和无监督的学习之间，因为同时使用标记和未标记的数据进行训练，通常是少量标记数据和大量未标记数据。使用这种方法的系统能够大大提高学习准确性。通常，当获取的标记数据需要熟练且相关的资源以对其进行训练/从中学习时，将选择半监督学习。

除了上述不同的学习算法，还有一种基于人工神经网络的学习算法，即深度学习。深度学习是机器学习的子集，通过使用大量标记数据和神经网络体系结构进行训练，这些神经网络体系结构可直接从数据中学习特征，而无需人工提取特征。深层神经网络最流行的类型之一是卷积神经网络（CNN或ConvNet）。

深度学习是机器学习的一种特殊形式。 机器学习工作流程首先从图像中手动提取相关功能。 然后使用这些特征创建一个模型，对图像中的对象进行分类。 通过深度学习工作流程，可以从图像中自动提取相关功能。 此外，深度学习执行“端到端学习”，即为网络提供原始数据和要执行的任务（例如分类），并自动学习如何执行此操作。

另一个关键差异是深度学习算法可随数据扩展，而浅层学习则收敛。 浅层学习是指在向网络添加更多示例和训练数据时达到一定性能水平的机器学习方法

深度学习网络的一个关键优势在于，随着数据规模的增加，它们通常会不断改善。

## 本章小结

# 链路泛洪攻击模型设计

## 链路图的建立

为了泛洪通往目标区域的链路，攻击者需要构建该区域周围的Internet的链路图。攻击者选择某个地理区域作为目标，并构建其网络链路的地图（即链路地图）。攻击者从僵尸机到目标区域内和周围的公共服务器运行多个跟踪路由，收集所有结果并对其进行处理以构建链路图。由于获得的某些路由可能不稳定，因此无法直接使用获得的链路图来查找目标链路。路由不稳定会使攻击复杂化，因为攻击者最终可能会追赶移动的目标。路由不稳定主要是由ISP的负载均衡过程（即通过多条路由转发流量）引起的，大多数商用路由器都支持这种过程。负载平衡的结果是，对于相同的僵尸机到诱饵服务器对，某些链路并不总是出现在由traceroute的多次调用产生的路由的跟踪上。因此，攻击者会定期执行跟踪路由，以验证对其构造的链路图是否有任何修改。如果路径发生更改，攻击者将使用更新的数据进行设置并再次发起攻击。

路由探测工具最常见的有traceroute，原理是利用网络层的ICMP协议中的超时和端口不可达两种回复报文来记录到达目标主机路径的路由信息

本文中的网络拓扑是基于SDN和openflow协议的，不支持traceroute工具进行跟踪路由，这里直接访问SDN控制器北向接口来获取链路图。

## 攻击建立

### 计算流量密度

攻击者分析目标区域的链路图，并为链路图中的每个网络链路计算只针对目标区域的攻击流量密度。目标链路的流量密度定义为可以通过该链路在僵尸机和目标区域服务器之间创建流的数量。因此，流量密度是特定于目标区域的度量标准。链路的高流量密度表示该链路将大量攻击和合法流量传递到特定目标区域，因此该链路成为良好的攻击目标。根据[cfa],可知流量密度遵循链路图中的幂律分布，这使攻击者可以轻松发现一组高流量密度链路，这些链路将最多的流量传递到目标区域。 此外，由于众所周知的Internet路由的长期稳定性，计算出的流量密度在至少几个小时内基本保持不变[cfa27，21]。因此，在选择目标链路时，本文将流量密度用作稳定可靠的度量标准。（公式）

### 目标链路选择

在此步骤中，攻击者发现多个不相交的集合目标链路被淹没。攻击者选择至少两个不相交的目标链路集，并每个依次使用一次，以实现攻击持久性。此步骤的目标是通过使用链路图和流量密度来选择最优的目标链路，以最大程度地增加流入目标区域的中断流量。为了量化通过选定的目标链路集可以拦截到目标区域的流量，攻击者计算该目标区域的降级率。降级率是被攻击削减的机器人目标区域路由数量占所有可能的机器人目标区域路由数量的比例。这里被切断是指如果一条路线包含被攻击淹没的目标链路，那么该路线就会被攻击切断。要选择最大化目标区域降级率的目标链路，攻击者必须解决广义的最大覆盖率问题，即是一个众所周知的NP难题，本文使用了有效的启发式方法，即贪心算法。本文的启发式方法的执行时间非常短，即在所有实验中都少于一分钟。如果需要的话，这可使攻击者适应动态的路由变化。

## 僵尸机的协调

一旦选择了目标链路（图2），攻击者就会协调各个僵尸机泛洪目标链路。 为了为给定的一组目标链路创建泛洪流，攻击者为每个僵尸机分配诱饵服务器列表和目的地为各个诱饵服务器的数据包的发送速率。发送速率的分配方式使单个攻击流的强度较低（或带宽较低），而其总带宽能够泛洪所有目标链路。此步骤包含下面两个子步骤。

### 攻击流的分配

攻击流分配的目标是使每个目标链路的总流量率略高于链路带宽，以便所有合法流量在这些链路中受到严重影响。分配策略必须满足两个约束：一是攻击者必须将每个流量保持在足够低的水平，以使服务器和附近的路由器以及入侵检测系统（IDS）中的任何网络保护机制都无法将流识别为恶意流。二是泛洪所有目标链路所需的总攻击流量相对平均地分配给了多个僵尸机和诱饵服务器。第一个约束条件确保了攻击流的不可区分性，而第二个约束条件则解决了目标区域中的服务器和诱饵服务器中的不可检测性。攻击者首先为每个目标链路设置最大目标带宽，并用攻击流将其耗尽。然后，为每个目标链路分配单独的流。

在目标链路上的每条攻击流的发送速率受流量密度限制。目标链路的每条流的平均速率应高于目标带宽除以链路上可用攻击流的最大数量，该数量与其流密度成正比。此外，按流量的分配必须考虑诱饵服务器在不触发流量预警的情况下可以处理的最大流量。例如，如果诱饵服务器是公共Web服务器，则平均每秒钟Web点击一次（每秒HTTP GET数据包4 Kbps）将不会被分类为服务器上的异常流量。因此，攻击者可以轻松地分配足够多的攻击流，且每流速率较低。为了增强僵尸程序和诱饵服务器上攻击流量的不可检测性，每当分配新的攻击流时，攻击者都必须考虑僵尸程序和诱饵服务器上所有以前分配的流量速率。尽管攻击者进行了精心的攻击流程分配，但位于目标链路上游的非目标链路如果带宽有限或某个区域的僵尸机密度过高，则仍可能会被拥塞，本文称之为早期拥塞。攻击者可以使用先验链路带宽估计避免潜在的早期拥塞。

### 目标链路泛洪

攻击者指示她的机器人开始生成攻击流。 每个僵尸机负责多个攻击流，每个攻击流都被分配一个具有相应所需发送速率的独特诱饵服务器。 僵尸程序会慢慢将其攻击流的发送速率提高到指定的发送速率，这使得攻击流与典型的“闪存人群”的流量模式【cfa29】无法区分。 僵尸主机可以根据每个目标链路的状态动态调整其流量强度。 即，如果目标链路的实际带宽小于分配的攻击带宽（在第II-C1节中设置），则一旦目标链路被泛洪，僵尸程序就会停止增加攻击流量。

为了避免了早期拥塞，即位于目标链路上游的非目标链路或更多链路变得拥塞的事件。本文认为，早期的拥塞不会影响攻击的可行性，而是攻击提供的问题，这是一个非常独特且易于被攻击者处理的问题。僵尸机可以通过定期执行到目标区域的跟踪路由来轻松检测早期拥塞，因为如果发生这种情况，它们将不会从拥挤的路由器和该路由上的后续路由器收到大多数答复（即，ICMP回复超时）。当僵尸机报告早期拥塞时，攻击者可以将攻击流重新分配给其他僵尸机，以避免早期拥塞。除了动态分配攻击流之外，攻击者还可以指示僵尸机使用先验带宽估算工具（例如Pathneck [40]）估算出指向目标链路的可用带宽，并在分配攻击流之前预测早期拥塞。这样，攻击者就可以配置僵尸机，从而避免早期拥塞。

## 动态攻击

攻击者可以动态地更改目标链路的集合（在先前找到的多个集合中），并无限期地延长Crossfire攻击的持续时间。同一组目标链路的连续链路泛洪将导致僵尸机到诱饵服务器路由发生变动，因为泛洪攻击会不可避免地激活路由器的故障检测机制。因此，更改目标链路的集合可确保攻击持久性并使攻击保持纯数据平面攻击。攻击者还可以动态更改僵尸主机集，以进一步增强Crossfire攻击的不可检测性。

如果攻击者动态地更改攻击参数（例如，僵尸机，诱饵服务器和目标链路），同时又保持相同的目标区域，那么LFA就会持续进行。攻击者可以采用动态攻击，以进一步提高攻击流量与合法流量的不可区分性，以及按目标区域对目标链路的不可检测性。根据可以动态更改的攻击参数的类型，本文动态攻击可以分为两种：一种在保持相同目标链路的同时更改僵尸机和诱饵服务器，另一种是更改目标链路，同时保持相同的目标区域。

第一种攻击类型的主要优点是，它在保持相同攻击效果的同时，进一步提高了攻击流和合法流的不可区分性。由于在选定目标链路上看到的源IP地址和目标IP地址会随时间变化，因此ISP无法轻松识别造成攻击的源IP地址和目标IP地址。潜在的缺点是，这种攻击需要比泛洪目标链路所需的最低数量更多的僵尸机和诱饵服务器。但是，目前的僵尸机成本表明这种缺点微不足道。

第二种动态攻击对同一目标区域使用多组不相交的目标链路。为了找到多个不相交的集合，攻击者依次执行目标链路选择算法。即，在删除先前选择的链路之后，选择n个最佳的目标链路集。由于ISP无法确定攻击者选择的目标链路，因此使用多个不相交的目标链路集会增强ISP无法检测到攻击的可能性。更重要的是，如前一部分所述，这种类型的动态攻击使LFA保持纯数据平面攻击。潜在的缺点是，这种类型的动态攻击可能会降低LFA的效果，因为攻击非最佳目标链路集导致的降级率可能低于攻击最佳集合的降级率。为了在未被发现的情况下最大程度地提高攻击效果，攻击者可以改变目标链路集。攻击者可以在大多数攻击中使用最佳链路集合，并仅在很短的时间间隔内切换到非最佳链路集合。例如，如果攻击者反复安排3分钟进行最佳链路集合攻击，然后再安排30秒进行第二最佳链路集合攻击，则可以长期地维持对目标区域的攻击，同时将降级率的降低限制在4％以下。

## 本章小结

# 基于SDN的链路泛洪攻击检测方法

## 基于SDN的链路选择

### 

## 数据重放和采集

### 发

## 特征提取

## 基于LSTM模型的LFA检测方法

### LSTM算法简介

## 基于CNN模型的LFA检测方法

### CNN算法简介

# 实验设计与分析

本文第3章和第4章分别详细介绍了LFA模型的实现和基于SDN的LFA检测方法。本章将在上述两章的基础上通过mininet网络仿真来搭建实验环境，然后介绍LFA模型的实现并证明其有效性，最后通过实验验证第4章检测方法的实用效果。

## 实验平台

为了验证LFA模型的有效性以及本文提出的LFA检测方法的实验结果，本文利用Mininet虚拟网络仿真工具，并使用了OpenvSwitch虚拟交换机，结合Opendaylight控制器，搭建了实验环境。

### Mininet

Mininet是一个网络模拟器，可以创建虚拟主机，交换机，控制器和网络链路。 Mininet主机运行在Linux环境下，其交换机支持OpenFlow，以实现高度灵活的自定义路由和软件定义的网络。Mininet支持可重复且易于打包的系统级回归测试，无需复杂的物理连接即可进行复杂的拓扑测试。Mininet包括一个命令行接口，用于调试或运行网络范围的测试，xterm可以支持对Mininet主机进一步的测试。支持任意的自定义拓扑，并包括一组基本的参数化拓扑，无需编程即可使用，而且提供了简单明了的用于网络创建和实验的可扩展Python API。 Mininet使用基于进程的虚拟化技术在单个操作系统内核上运行多个主机和交换机。从2.2.26版开始，Linux内核支持网络命名空间技术，是一种轻量级的虚拟化功能，可为单独的进程提供单独的网络接口，路由表和ARP表，即对网络资源的隔离能力。Mininet使用网络命名空间技术来创建虚拟主机host，使得不同host拥有各自独立的网络命名空间。本文在Mininet上搭建了网络拓扑如图5.2所示。

### OVS

Openvswitch(OVS)作为一个软件层面的虚拟交换机，OVS架构如图5.3所示，主要包括内核空间的datapath和用户空间的vswitchd进程，ovsdb等组件。其中，内核模块datapath主要从网卡读取数据包，并跟流表中的流表项进行匹配，成功匹配后立即转发，否则交给vswitchd进程处理。datapath在端口绑定和初始化时注册hook函数，把端口的数据包交给内核处理。vswitchd是一个守护进程，使用OpenFlow与外界进行通信，使用OVSDB协议与ovsdb服务器进行通信，通过netlink与内核进行通信，通过netdev抽象接口与系统进行通信；ovsdb主要是保存了ovs配置信息的数据库。

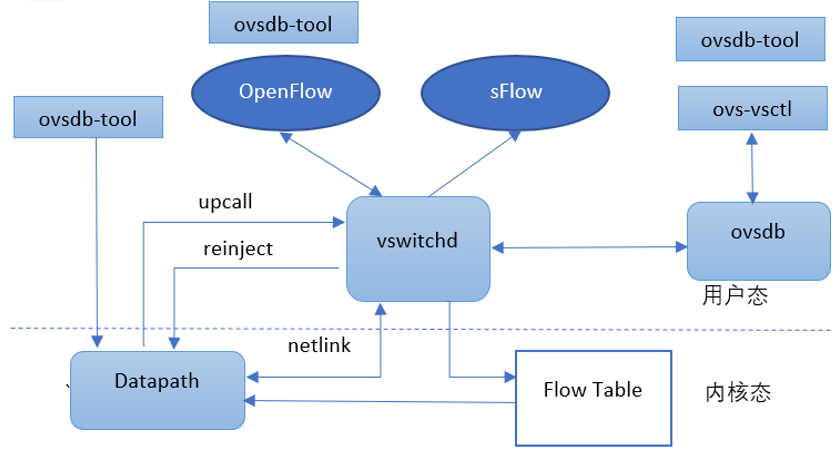


图5.1 OVS架构图

### Opendaylight控制器

OpenDaylight是IBM，Cisco，Juniper，VMWare和其他几个主要网络供应商支持的开放源代码项目，已经依次发布了氢Hydrogen、氦Helium、锂Lithium、铍Beryllium、硼Boron、碳Carbon、氮Nitrogen、氧Oxygen 八个版本。OpenDaylight Controller仅通过软件实现，并保留在其自己的Java虚拟机（VM）（JVM）中。这意味着它可以部署在支持Java的硬件和任何操作系统平台上。Opendaylight控制器使用以下工具：Maven用于构建自动化；OSGi用于动态加载包和打包JAR文件；JAVA接口用于事件侦听，规范和形成模式。

OpenDaylight控制器的体系结构如图5.1所示，它可以部署在支持Java的任何硬件和操作系统平台上。OpenDaylight控制器具有多层，从图可以看出跟SDN架构基本一致。顶层由业务和网络逻辑应用程序组成，中间层是框架层，底层由物理和虚拟设备组成。中间层是可以在其中体现SDN抽象的框架。该层托管北向和南向API。OpenDaylight控制器公开了应用程序使用的开放式北向API，支持OSGi框架和北向API的双向REST。业务逻辑驻留在中间层之上的应用程序中，OpenDaylight控制器公开了开放的北向API，供应用程序使用。这些应用程序使用控制器收集有关网络的信息，运行算法进行分析，然后使用OpenDaylight控制器在整个网络中创建新规则。南向接口能够作为单独的插件支持多种协议，例如OpenFlow和BGP-LS。SAL决定如何实现请求的服务，而与控制器和网络设备之间使用的基础协议无关。

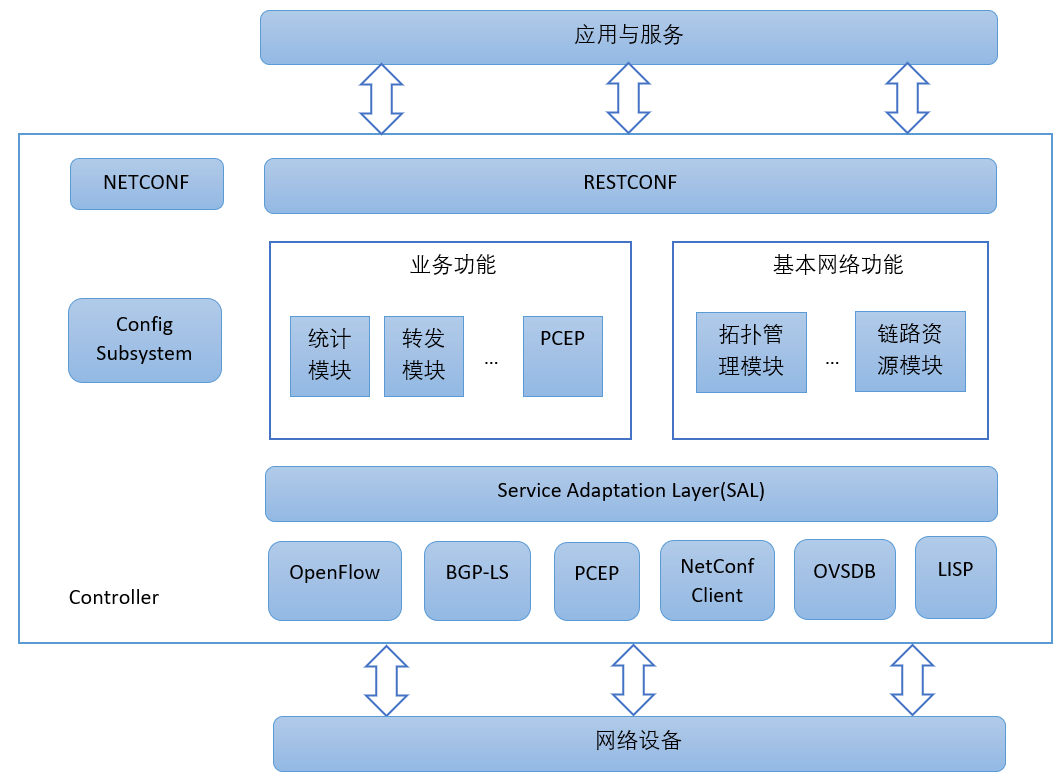


图 5.1 Opendaylight框架示意图

自OpenDaylight控制器的第一个版本Hydrogen发布以来，OpenDaylight经历了使用Apache Karaf容器，具有新的用户界面以及更简化和可定制的安装过程与OpenStack进行了更深入的集成，包括对OVSDB Integration项目的改进以及其他功能，如安全组，分布式虚拟路由器和负载平衡即服务，实现对YANG模型的支持等过程。直到最近的一个版本Oxygen支持了P4插件，以在OpenDaylight平台中包括数据平面抽象，还包括一个Kubernetes插件来支持基于容器的工作负载，以及一个用于混合VM容器环境的Neutron Northbound扩展。本文选择了已经相对稳定，功能更多的Oxygen版本作为网络拓扑的控制器。

## LFA模型的实验测试

### 链路图的建立

根据第3章，攻击者在LFA发动前需要通过路由跟踪来得到僵尸机到诱饵服务器之间的转发路径，在传统网络中可以通过运行traceroute工具来实现，由于这里使用的是基于openvswitch虚拟交换机不支持第三层协议，无法通过traceroute得到ICMP数据包time exceeded消息报文，这里直接通过Opendaylight控制器提供的北向接口可以很容易得到网络的链路信息取得跟最终实验一致的结果。

链路图的建立模块是使用python实现的，通过Opendaylight的REST API获取链路信息，对控制器发送get请求，可以返回json数据然后用字典保存所有链路信息。通过Yang Man访问“restconf/operational/network-topology:network-topology”目录下的API就可以得到网络拓扑中所有的链路信息。

### 目标链路选择实现

通过上一步得到链路信息后，执行第3章的目标链路选择算法，这里访问Opendaylight控制器的REST API，主要API放在“restconf/operational/opendaylight-inventory:nodes”目录下，通过发送get请求可以得到网络所有流实时数据，解析返回的json数据便可得到所有链路经过的所有流信息，根据前面章节的算法便可以统计出每条流源ip和目的ip的地址空间，并算出每条链路的权重，这里使用python把链路和对应权重用字典保存，执行链路选择算法得到目标链路。由于结果返回的json数据比较复杂，这里用多个字典保存了中间结果然后多次迭代进行归并到一个字典上来。

### 路由转发策略

为了仿真现实网络，这里通过Opendaylight控制器用java实现了自定义的路由转发策略。通过基于最小跳数路由，来自定义新的转发模块。这里的最小跳数路由是指交换机根据目的地址对应的网段，选择一条到目的地址网段的最短跳数路径转发。（流程图）

主要思路如下，以交换机S1到S2为例，首先控制器收到交换机S1上发送的packetIn(主机h1到h2)消息；然后查hostMac-switchPort表，确定目的交换机s2；根据全局拓扑视图，确定从s1到s2的最小跳数路径；最后发送packetOut消息并下发流表指示之后流从哪个端口经过。

主要流程：

注册成为packetIn的监听器，监听IP packetIn消息

收包后查hostMac-switchPort表，找到目的交换机

根据本地拓扑结构，确定源交换机到目的交换机的最短跳数路径

找到位于最短路上的出端口后，给交换机下发packetOut和addflow

关键数据结构：

a）hostMac-switchPort表 、 本地拓扑结构

b）本地拓扑是基于Jung的图。

c）注册DataStore的监听器，监听交换机链路变化事件，根据收到的链路变化更新本地拓扑结构和hostMac-switchPort表。

### 僵尸网络模型

根据图5.2的网络拓扑可知，我们可以通过上面步骤中的链路选择算法选择可以泛洪目标链路的源主机和目的主机，在这里分别对应为僵尸机和诱饵服务器。

### 流量模型

为了满足仿真网络的需求，需要建立一个能够反映实际流量特点并对本实验中的LFA模型进行有效验证的背景流量模型。

本文中带宽模型直接使用了固定带宽，这样在发生阻塞时可以测试到实验结果。攻击流量的生成主要使用iperf工具依次发送不同速率的数据包，实验过程使用了链路带宽总共达到10Mbps到120Mbps进行测试。这里背景流量使用了最广泛使用的泊松模型。泊松分布是用于分析传统统电话网络流量的主要模型，泊松模型主要指在时间序列内，数据包到达的数量符合参数为的泊松分布，用公式表示如下：

(1)

这里，参数表示单位时间内到达的数据包期望值也就是包到达的平均速率，并满足。如果到达来自大量独立来源（称为泊松来源），则泊松分布是适当的，这与本文实验环境相符。

## LFA检测的实验结果

数据重放，使用IDS数据集

# 总结与展望

## 本文工作总结

## 未来工作展望

参考文献

[1] 杨瑞林, 李力军. 新型低合金高强韧性耐磨钢的研究. 钢铁. 1999（7）：41~45

[2] Schinstock, D.E., Cuttino, J.F. Real time kinematic solutions of a non-contacting, three dimensional metrology frame[J]. Precision Engineering. 2000, 24(1):70-76

[3] 温诗铸. 摩擦学原理. 北京：清华大学出版社. 1990：296-300

[4] 贾名字. 工程硕士论文撰写规范[硕士论文].上海：上海交通大学. 2000

攻读硕士学位期间主要的研究成果

[1] Schinstock, D.E., Cuttino, J.F. Real time kinematic solutions of a non-contacting, three dimensional metrology frame[J]. Precision Engineering. 2000, 24(1):70-76

[2] Schinstock, D.E., Cuttino, J.F. Real time kinematic solutions of a non-contacting, three dimensional metrology frame[J]. Precision Engineering. 2000, 24(1):70-76

致谢