封面

摘要（中文）关键词

摘要（英文）关键词

目录：

1. 绪论

1.1研究背景

1.2国内外研究现状及趋势

1.3论文研究意义及创新

1.4论文结构安排

2、数据中心拥塞控制基础

2.1数据中心网络模型基础

2.2拥塞控制算法及机制

2.2.1流量调度

2.2.2负载均衡

2.2.3发送速率控制

2.3算法评价分析

2.4本章小结

3.基于神经网络的拥塞控制算法研究

3.1算法整体设计及模块划分

3.2数据预处理模块设计

3.2.1算法设计

3.2.2算法分析验证

3.3RTT预测模块设计

3.3.1神经网络模型设计

3.3.2数据集、损失函数、优化器等参数设计

3.3.3算法分析验证

3.4PID速率控制模块设计

3.4.1网络模型搭建

3.4.2参数设计

3.4.3模型分析

3.5本章小结

4、神经网络模型训练及仿真展示

4.1 NS3仿真平台环境搭建及设计

4.1.1环境配置

4.1.2网络拓扑仿真设计

4.2神经网络模块化训练

4.2.1数据集获取及展示

4.2.2RTT预测模块训练效果展示

4.2.3PID速率控制模块训练效果展示

4.3算法拥塞控制效果展示

4.3.1RTT时延

4.3.2发送速率

4.4本章小结

5.算法分析比较

5.0分析比较指标

5.1算法比较

5.1.1TIMELY

5.1.2HPCC

5.1.3DCTCP

5.2比较结果分析总结

5.2.1算法优势及缺点

5.2.2算法创新点

6、全文总结及期望

6.1全文总结

6.2后续工作展望

致谢

参考文献

摘要:数据中心网络传输旨在实现低延时和无损行为。其中，RTT作为包往返时间，被证明和交换机队列长度有着很大的关联程度，是网络终端节点能够获取的为数不多的拥塞信号。目前随着网络硬件的进步，RTT的测量已经能够达到微妙级的精度。然后，我们设计了一种神经网络拥塞控制算法，充分利用了数据包RTT时延，来对发送速率进行调整，保持低延时和高带宽。我们在基于NS3搭建的仿真平台上部署了该算法。设计容易发生拥塞的INCAST通信模式，通过实验表明，它提供了很好的拥塞控制效果，平均时延控制在10us以下，在大规模数据注入时也没有发生拥塞。同时该算法不依赖于网络硬件的修改，在数据中心等网络大规模部署是十分容易的。

关键词：神经网络、拥塞控制、数据中心网络、INCAST、RTT

Abstract: Data center network transmissions are designed for low latency and lossless behavior. Among them, RTT, as the packet round-trip time, proves to have a great correlation with the length of the switch queue, and is one of the few congestion signals that network endpoints can obtain. At present, with the advancement of network hardware, RTT measurements have been able to achieve subtle levels of accuracy. Then, we design a neural network congestion control algorithm that makes full use of the packet RTT delay to adjust the transmission rate to keep the latency low and high bandwidth. We deployed the algorithm on a simulation platform based on NS3. The INCAST communication mode is designed to be prone to congestion, and experiments show that it provides good congestion control effect, the average delay is controlled below 10us, and no congestion occurs during large-scale data injection. At the same time, the algorithm does not depend on the modification of network hardware, and it is very easy to deploy it on a large scale in networks such as data centers.

Keywords: neural networks, congestion control, data center networks, INCAST, RTT

1. 绪论

1.1研究背景

根据2021年颁布的《数据中心智能无损网络白皮书》[1]，数据中心支持了很多类型的业务，包括大数据机器学习等；随着人工智能在数据中心的发展和应用，数据中心中传输的数据呈指数级的增长，海量数据的传送使网络承受更大的传输压力，数据发送的瓶颈逐渐从终端转移到网络设备上，低时延和无损行为成为了数据中心网络新的必要需求。

云时代的数据中心专注于应用转型和服务的快速部署。在 AI 时代，数据中 心提供了实现数字化生活所需的信息和算法。高速存储和人工智能分布式计算的 结合，将大数据转化为快速数据，供人、机、物访问。高性能、大规模、无丢包 的数据中心网络对数字转换的顺利进行至关重要。

人工智能、网络性能等高性能应用的关键指标包括吞吐量、时延和拥塞。吞 吐量是指快速传输大量数据的网络总容量。时延是指跨数据中心网络事务的总延迟。当流量超过网络容量时，会发生拥塞。丢包是严重影响吞吐量和时延的因素。

为应对这种变化趋势，数据中心使用了新远程直接内存访问（Remote Direct Memory Access, RDMA） 协议消除了数据副本，释放了 CPU 资源，能够完成路径选择和取出顺序计算，但是RDMA 效率的提高给网络带来了更大的压力，将瓶颈转移到数据中心网络基础设施上，这使得网络比以往更容易发生拥塞，而传统拥塞控制算法已经无法适应目前的数据中心数据传输，因此针对数据中心网络设计新的拥塞控制算法和机制就非常重要。

1.2研究现状及趋势

数据中心无损网络的拥塞控制，目前已提出了多种解决思路和方法，大体方向一共有三种，分别是仅通过端节点的拥塞控制算法、需要网络中交换机或控制器配合的流调度和负载均衡；通过端节点的拥塞控制算法又包含了基于丢包的TCP Cubic算法、基于ECN丢包的DCQCN[2]和DCTCP[3]算法、基于RTT信息的TIMELY[4]算法等，流调度包括了基于截止时间的D2TCP[5]、基于流大小的pFabric[6]等，负载均衡包括了基础简单算法ECMP和复杂算法Hedera[7].大部分拥塞控制算法由于需要特殊的硬件或者报文支持，如HPCC[8]算法需要INT报文字段，阿里提出的vFabric[9]需要能够发送特殊探针的边界路由器。

拥塞控制研究发展有以下几个方向，一是基于SDN的中心化控制，集中式控制可以对网络全局信息进行掌控，更能精确的规划控制流量，但同时面临着可部署规模的问题；二是人工智能与流量控制技术结合的研究方向，人工智能具有强大的自适应力和自学能力，能为拥塞控制提供一套有效的决策工具[10]，例如华为在2021年提出的ACC[11]中提出了强化学习的拥塞控制算法等。

1.3论文研究意义及创新

由于数据中心网络的数据传送的突发性、以及流种类的复杂性，目前的拥塞控制算法和机制主要有两个特点（缺点）：一是具有针对性，对于特定情况或特定流的拥塞处理较好，但是对于网络中的突发流量导致的拥塞情况效果较差；二是大部分算法都需要专门的硬件配合来获取所需的信息，但数据中心网络规模非常大，大部分未实现商用。

本论文基于数据中心提出了一种新的拥塞控制算法，结合了AI人工智能，设计了一个新的LSTM+PIDNN神经网络框架。解决了 上述两个缺点，首先神经网络具有很强的学习能力，能够根据网络状态变化调整参数，解决了大部分算法的“针对性”；同时神经网络的实现以及相应数据（RTT）的获取不需要通过专门的硬件实现，而只需要相应的软件修改；因此更容易大规模部署；

1.4论文结构安排

本文的章节安排如下：

第一章阐述了数据中心网络的背景及拥塞控制算法概况。对当前数据中心中拥塞控制算法进行了总结了分析，然后介绍了本论文的算法设计思想。

第二章讲述了数据中心网络的拓扑及通信模式的基本概念，以及不同拥塞控制算法的基本概念。包括对于CLOS拓扑和INCAST通信模式，以及DCTCP、TIMELY、HCPP、ECMP等。

第三章介绍了本论文设计的基于神经网络的数据中心拥塞控制算法，包括了模块划分，模型设计搭建及分析验证，以及数据集设计和网络训练方式。

第四章介绍了算法仿真效果。包括网络拓扑搭建及模型训练，效果分析。

第五章对算法效果和其他拥塞控制算法进行比较，主要指标包括平均RTT、发送速率等。

第六章对全文进行了总结。

2、数据中心拥塞控制基础

2.1数据中心网络模型基础

2.1.1数据中心网络拓扑

新一代数据中心网络架构大多数都为CLOS网络架构，CLOS是一个对称的“接入-核心-接入”网络架构，具有以下特点

* ·多级交换，典型为三级交换架构。
* 在每一级的每个单元都与下一级的设备全连接。
* 到指定目的地，在第1级交换单元存在多条路由，而后续交换单元都只存在唯一的一条路由。
* 严格意义上的无阻塞。
* 支持递归，可无限扩展。



目前，基于CLOS架构搭建的数据中心网络拓扑有Fat-Tree和Spine-Leaf

2.1.2数据中心网络协议

高带宽、低延迟是目前数据中心应用的基本需求。RDMA 通过 Memory Region 机制使得网卡能够直接读写用户态的内存数据，避免了数据拷贝和上下文切换；并将网络协议栈从软件实现 offload 到网卡硬件实现，极大降低了 CPU 开销。随着 RoCEv2（RDMA over Converged Ethernet v2）技术的成熟，RDMA 可以部署在数据中心已有的网络设施上，RDMA 成为数据中心高速网络通信的主流方案。



使用RDMA协议有以下优点：

* 零拷贝(Zero-copy)
* 内核旁路(Kernel bypass)
* 不需要CPU干预(No CPU involvement消息基于事务(Message based transactions)
* 支持分散/聚合条目(Scatter/gather entries support)

2.1.3数据中心网络对于拥塞控制协议的要求

由于RDMA的重传机制，丢包对 RDMA 造成大幅性能下降，因此 RDMA 要求 L2 层或者 L3 层网络是无损的。



丢包对 RDMA 吞吐量的影响

由上图可知，RDMA吞吐量对于丢包率的增长是十分敏感的，当网络丢包率到1%,时，RDMA的读写速率下降到几乎为1零，只有当丢包率保持在一个非常低的比率时，才能够完整发挥RDMA的性能；

为了实现不丢包，数据中心网络RoCEv2引入了基于优先级的流量控制PFC协议，基于优先级的流量控制（PFC）功能是当接收设备输入缓存区的占用超过设 定阈值时，暂停上游发送设备，防止因缓存区溢出造成丢包。虽然提供了 RoCE 所必需的无损环境，但 PFC 的大规模使用也存在一些问题，包括 PFC 死锁的可能性。

因此，单是上述的PFC协议是无法避免网络拥塞的，数据中心网络还需在网络上的各个节点采取不同的拥塞控制协议或者机制，多者之间联合调控。

2.2拥塞控制算法及机制

根据算法执行的位置进行分类，拥塞控制算法主要分为在终端节点控制的拥塞控制协议和在交换机等交换设备上实现的流量工程；两者虽然实现在CDN网络中不同的位置节点，但是都是为了合理调度网络资源以减少网络拥塞情况的发生。下面将具体进行介绍

2.2.1拥塞控制

在终端节点是拥塞控制主要就是根据获取的网络信息来控制TCP层发送速率，网络信息大多数从接收到的数据包中某个标志字段分析得出，大致分为三类：基于丢包率（TCP cubic）、基于ECN标志字段(DCTCP,HULL,DCQCN)和RTT(TIMELY);

2.2.2流量工程

流量工程主要是实现在网络中交换机等交换设备上，主要分为流调度和负载均衡两种；流调度以流为调度单位，主要在端口队列实现。通过某种机制或算法控制流的发送数量或顺序，有基于截止时间的D2TCP、基于流大小的pFabric和基于链路价格的FCP等；而复杂均衡则是控制流或数据包走不同的网络链路，避免大量数据占用同一条链路资源，使得不同物理路径之间负载平衡，例如ECMP算法；

除了拥塞控制和流量工程之外，有些拥塞控制算法结合了二者；例如阿里巴巴提出uFAB算法，需要终端和边缘交换机联合调控，边缘交换机发送探测数据包收集网络信息，后再将统计信息发送给连接的终端节点，终端节点根据网络状态控制流发送速率；同时交换机还可以根据网络状态控制流发送路径。



2.3算法评价分析

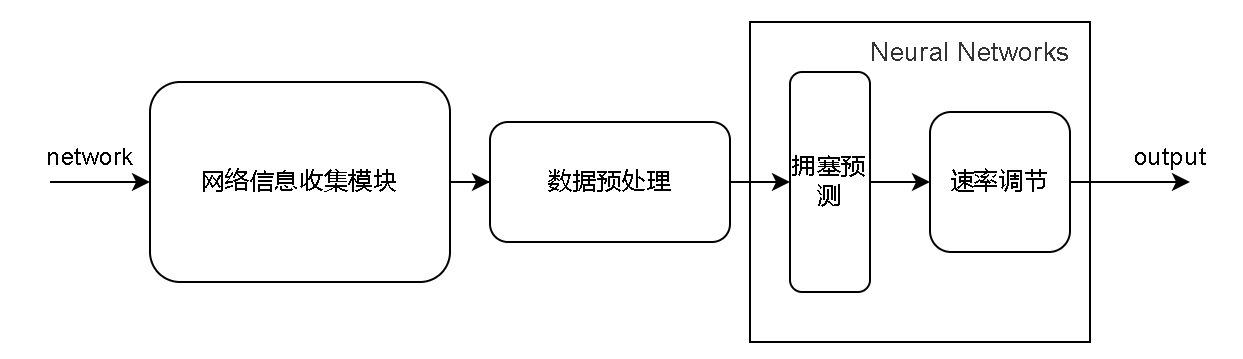
目前数据中心拥塞控制算法种类繁多；其中大部分对于特定拥塞场景或者对于特定东西向流（视频流、语音流、AI流等）有着很好的控制效果，但是不能够作为处理拥塞的普适算法；其中一些算法需要对网络协议中报文字段进行增加或者对设备硬件进行修改，这些算法在拥塞处理上结果很理想，能够达到很好的预期，但是往往不能够大规模部署，停留在实验阶段。很多拥塞算法参数固定，对网络状态变化适应能力差，速率控制比较机械，不够灵活，不能够适应智能化网络发展趋势。

2.4本章小结

本章先介绍了数据中心网络协议模型，再根据种类介绍了多种拥塞控制算法，并对他们的优缺点进行分析总结。

3.基于神经网络的拥塞控制算法研究

3.1算法整体设计及模块划分



3.2数据预处理模块设计

3.2.1算法设计

3.2.2算法分析验证

3.3RTT预测模块设计

3.3.1神经网络模型设计

3.3.2数据集、损失函数、优化器等参数设计

3.3.3算法分析验证

3.4PID速率控制模块设计

3.4.1网络模型搭建

3.4.2参数设计

3.4.3模型分析

3.5本章小结