数据中心网络关键技术项目研究进展报告

一、数据中心网络流量特征

1.1 细粒度的数据中心网络流量特征

分析了web、Cache、Hadoop三种业务的流量特征。在测试的过程中，数据中心使用的是传统的接入层、汇聚层和核心层的三层结构，每一个服务器都通过10Gbps的以太网链路与其所在rack上方的Tor-switch相连接，每一个Tor-switch相应地又与其上方的汇聚层交换机（”fabric”）通过40Gbps或100Gbps的链路相连接，然后，汇聚层交换机再与核心层交换机（”spines”）相连接，由此构成树状结构。本实验一共测量了30个rack上Tor-switch层在24小时内的流量数据，每个业务各占其中10个rack，并且每个rack上的所有服务器都只运行同一种业务，以确保采集到的数据能够代表相应业务的流量特征。实验取样的是所有rack上的Tor-switch连接的任意一个服务器端口的数据，每个小时任取其中的两分钟的数据进行记录，通过一个月的采集得到了720个这样的记录。

1.1.1 细粒度的流量分布特征

先对测量的数据以25µs的时间间隔进行采样，如果在采样间隔里链路的利用率超过了50%则我们称其在该时间间隔为Hot状态，如果一个链路在连续的采样间隔中出现了不间断的Hot状态则我们称其为一次burst。以下是关于Web、Cache、Hadoop三种业务burst的持续时间长度的CDF曲线，如图1所示：

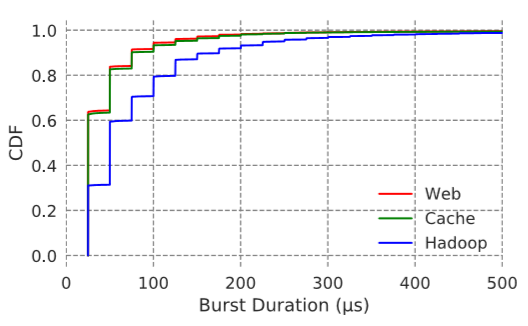


图1 .突发的持续时间分布

由持续时间的CDF图线可以发现，三种业务超过90%的burst的持续时间长度都小于200µs，而且Web业务对应rack的测量结果表明约90%的burst持续时间都要小于50µs，由此可以得出结论：链路处在高利用率的持续时间长度一般都很短，大概在200µs以内。

继续以25µs为时间间隔，再考虑相邻两个时间间隔是否出现burst是不是会相互影响。

令随机变量来表示在第t个时间间隔内是否出现Hot状态，表示第t个时间间隔内出现了Hot状态，表示第t个时间间隔内没有出现Hot状态。于是根据测得的结果得到以下数据，如图2所示：

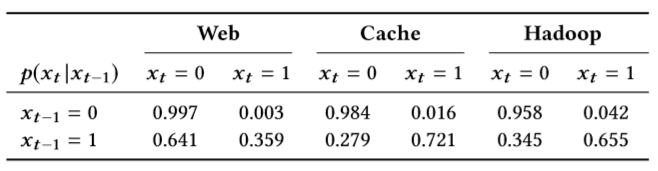


图2 突发间隔间的相关性

通过表格数据我们可以计算参数r=，如果则说明相邻两个间隔是否为burst状态没有相互影响，若或者则说明相邻两个间隔有联系。计算上述数据可得：

因为三种业务的，所以三种业务下，相邻时间间隔出现高利用率burst状态是相互影响的。

最后研究的是相邻两次burst之间的时间间隔大小规律，根据测得的数据可以得到25µs采样数据下相邻两次burst的时间间隔大小的CDF曲线，如图3所示：

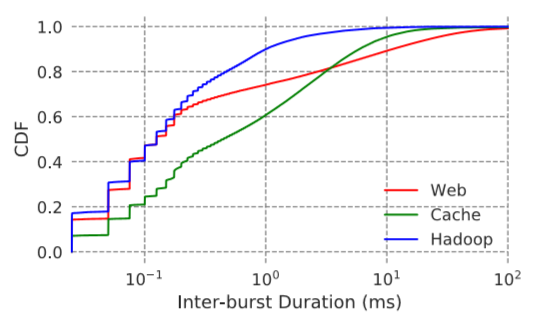


图3 突发间隔分布

根据曲线可以发现，相邻两次burst之间的时间间隔大多时候都是很小的，特别是对于Web业务和Cache业务来说，它们的间隔大小超过40%都是在0.1ms以下，但是之后的时间间隔大小的分布非常广，且随着时间间隔的增大其概率密度也不断变小。

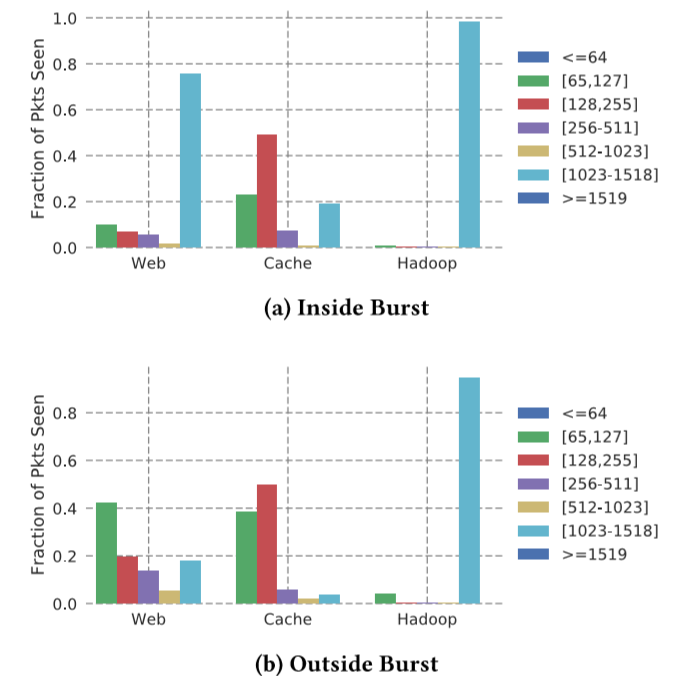


图4 三种业务的数据包大小分布

1.1.2数据包大小特征

根据所测得的数据，得到在burst中和burst以外Web、Cache、Hadoop三种业务的数据包大小情况柱状图，如图4所示：

对比两张柱状图可以发现，大体上，burst以外的数据包大小要小于burst内的数据包，而在Hadoop业务下无论是burst内还是burst外主要都是大数据包（大概这与Hadoop业务本身的特点相关），对于Web业务来说在burst以内大数据包占主导，超过了数据包总量的75%，而在burst以外由中小尺寸的数据包主导，但对于Cache业务来说无论是burst以外还是burst内都是小数据包占大多数。

1.1.3 链路负载均衡特征

在本测量环境中，每个Tor-switch都分别通过四个上行和下行的链路与汇聚层交换机相连，Tor-switch采用Equal-Cost MultiPath（ECMP）的策略将负载分配至每一个链路上，原则上，这样的策略可以保证比较好的均衡效果，但是实际测试效果如下，用平均绝对离差（Mean Absolute Deviation）的CDF曲线表示，实线表示采样间隔为25µs的情况，虚线表示采样间隔为1s的情况，如图5所示：

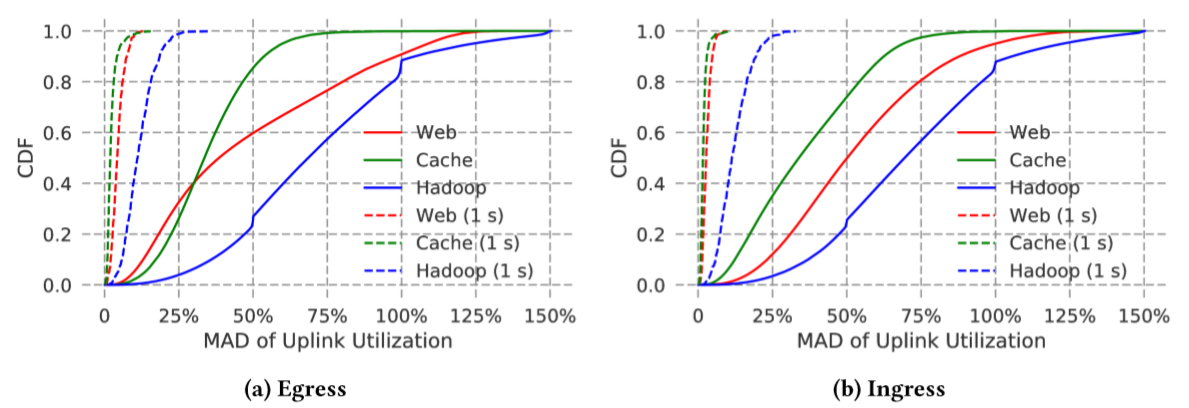


图5 链路利用率分布

有图可知在采样间隔为1s时，三种业务的负载相差并不大，基本做到了负载均衡，但是当用25µs的采样间隔进行采样时我们却发现，三种业务的负载都相差较大，特别是Hadoop业务，不同链路的负载情况甚至出现倍数关系，基本上可以判定从极短时间上看ECMP的负载均衡效果比较糟糕，其原因可能有两个：（1）ECMP的均衡对象是流而非数据包；（2）它使用的是五元组Hash，不能使均衡效果做到最优。

1.1.4 上行和下行链路繁忙情况

使用300µs采样周期进行测试，得到上行和下行链路在测试时间间隔中出现Hot状态的频数，并绘制以下上行和下行Hot状态比例柱状图：

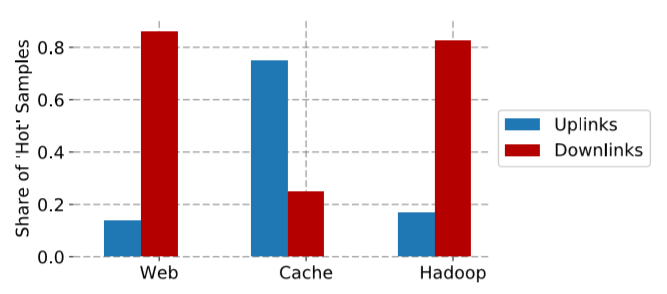


图6 上行和下行链路繁忙状态

根据图中信息，Web和Hadoop业务的Hot状态多出现于下行链路中，上行链路中出现的Hot状态的比例均不足20%。而Cache业务与之相反，在上行链路中更易出现Hot状态，而下行链路中出现Hot状态的比例不足30%。

1.1.5 Burst对Buffer占用率的影响

使用50ms的抽样周期对测量的流量数据抽样，测量发生burst的端口的比例与buffer占用率的关系，这里对buffer占用率根据其最大值进行归一化处理，得到以burst端口比例为横坐标、buffer归一化处理后的占用率为纵坐标的箱型图，如图7所示：

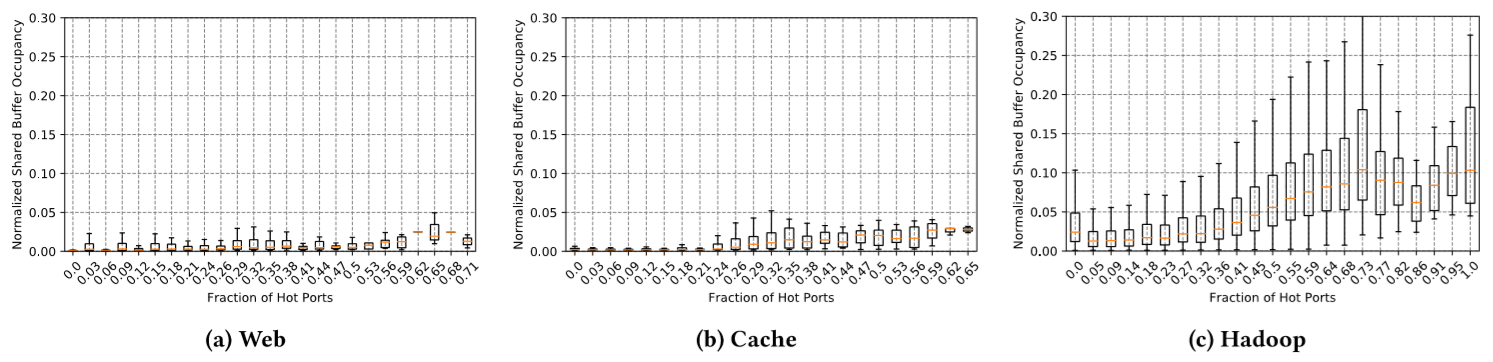


图7 Burst与端口buffer占用率

由图可知，Hadoop的burst的Hot状态端口的比例最大值最高，而Web和Cache业务处于高利用率的端口比例最高分别为71%和65%。而且相比于其他两种业务，Hadoop对buffer的占用率更高，而且占用率的分布范围也更广。同时，在burst端口比例较高时，三种业务的buffer占用率变化保持一致。

1.2 Facebook数据中心网络流量特征

1.2.1 Facebook数据中心网络拓扑结构

Facebook的数据中心分布在多个地理位置，不同位置的数据中心可以称作一个Datacenter Site，由骨干网络将所有的这些Site连接起来。而在每一个Datacenter Site中可能包含了一个或多个建筑，每个建筑都相当于是一个独立的数据中心，各自都运营着独立的数据中心网络。每个数据中心网络都有类似于图8所示的拓扑结构，它由多个cluster构成，cluster可是视作数据中心网络中部署的一个个单元，承载某个独立的业务，如hadoop、cache、frontend等，每个cluster都采用类似clos架构，即完全二分图结构，其中一层为top-of-rack switch（RSW），另一层为cluster switch（CSW），它们之间通过10Gbps的链路连接起来。每个RSW会连接很多承载特定服务的主机，他们通过10Gbps的以太网链路与对应的RSW相连接，每个主机都只运行单一的服务，同一RSW下的主机运行的服务都相同，同一个cluster下的主机运行的服务可能都相同，也可能不同，这具体取决于cluster本身承载的业务。每个RSW下连接的主机数量不固定，因cluster而异。不同cluster中的CSW通过称为Fat Cat的另一层交换机连接起来，单独的CSW和FC之间通过四条10Gbps的链路相连接，实现40Gbps的可用带宽。除此之外，CSW还通过连接汇聚层交换机负责与同一个site的不同数据中心网络进行数据传输，以及通过连接路由器实现不同site之间的数据传输。

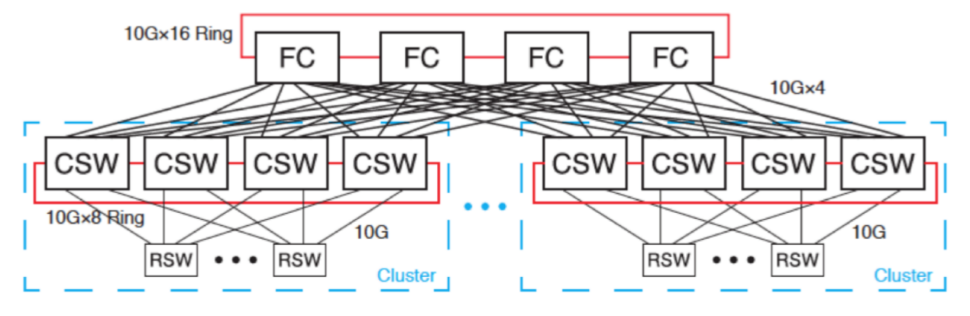


图8 Facebook数据中心网络拓扑结构

现今大多数的Facebook数据中心网络都还在采用4-post的clos架构，即每个数据中心网络中有4个FC，每个cluster中有4个CSW。而目前Facebook也已研发出新的Fabric架构，并开始在自己的数据中心中推广和应用。本文中所有的数据来源都是传统的图8所示结构。

1.2.2 业务概况

本文将考虑五种运行不同服务的主机，包括：运行web业务的web服务器，运行数据库业务的MySQL服务器，用于临时存储请求结果的cache服务器（既包括了保证数据一致性的leader cache，还包括了处理大多数读取请求的follower cache），进行离线数据分析和数据挖掘的hadoop服务器，以及聚合并供给信息资源的Multifeed服务器。这些服务涵盖了Facebook数据中心大多数的数据流量。

Facebook数据中心中还有少数服务类型不固定的主机，他们的服务类型会根据其目的动态调整。同时，Facebook数据中心也没有运用虚拟机，每一个服务都是运行在真实的物理主机上。

* + 1. 链路利用率特征

通过对服务器主机连接的10Gbps以太网链路测量可知，主机和他们对应的RSW之间的链路在1分钟内时间间隔的平均链路利用率不足1%，如此低的链路利用率大概与Facebook不久前将主机的接入层链路由1Gbps升级为了10Gbps有关。即使只关注负载最为严重的那些链路，99%的链路1分钟内的平均利用率也不足10%，而且不同cluster中的这些接入层链路负载也存在很大差别，负载最大的cluster（Hadoop业务）的平均链路利用率差不多是负载较小的cluster（Frontend）的5倍。

对于RSW和CSW之间的链路，从总体来看，链路利用率的中位数在10%-20%之间，若将cluster按负载大小排序，单看负载最为严重的前5%的那些cluster，链路利用率中位数在23%-46%之间，这个数据高出了之前大多数研究所得的结论。而不同cluster之间的差异相比于接入层来说减小了，负载最重的cluster的平均链路利用率是负载较轻的cluster的3倍。

而再往上层看，CSW和FC之间的链路利用率就更高了，而且不同cluster的差异也更小了，大概是因为不同cluster中的链路会根据其特定的需求进行过超额的分配，如cluster所有接入层链路的容量总和大大超出了该cluster与FC之间的链路容量，使得上层链路利用率较高，而且使得cluster之间的链路利用率差别不大。

* + 1. 流量传输的特征

通过以1秒为时间间隔，分别对Hadoop cluster的一个Hadoop服务器、Frontend cluster的一个web服务器和Cache cluster的一个cache follower和一个cache leader服务器的流量传输分布情况进行统计，得到了图2所示4台服务器的流量传输分布情况柱状图，图中通过四种不同颜色的柱形图案分别表示不同的流量传输分布情况，如蓝绿色柱形表示的是仅在同一个rack下的不同主机之间传输的流量大小，而深蓝色的柱形表示源宿主机在同一个cluster内但不在同一个rack下的情况，红色柱形表示源宿主机在同一个数据中心网络内但分布在不同的cluster中，绿色柱形表示源宿主机分布在不同数据中心网络中的情况。

由图9可知，Hadoop业务的流量分布情况变化最为剧烈，在某些时间间隔上有明显的流量传输过程，而还有一些时间间隔上没有明显的流量传输。从采集到的数据来看，该Hadoop服务器发送的99.8%的数据流量都发送给了其他的Hadoop服务器，其中，约75.7%的流量发往了该服务器所在同一rack下的其他服务器，而且该rack下的不同接收主机收到的流量大小差异并不大，换言之流量在该rack下的分配比较均匀。而剩下的几乎所有流量都发往了该cluster内其他rack的主机，只有极为少数的流量发送至了分布在该cluster外部的主机。从具体分布上看，发往该cluster不同rack的流量的接收方主机占该cluster中所有主机的1.5%，分布在该cluster内的95%的rack中，然而，其中80%的流量都发往了17%的rack，此结论与此前研究观测到的数据保持一致。而且，尽管hadoop服务器发送的流量基本发往了分布在同一个rack内或者同一个cluster内的主机，但具体这两者各占比多少却不断变化。Hadoop这样的多变性大概是其处理的job的大小和job所处的不同阶段共同导致的结果。

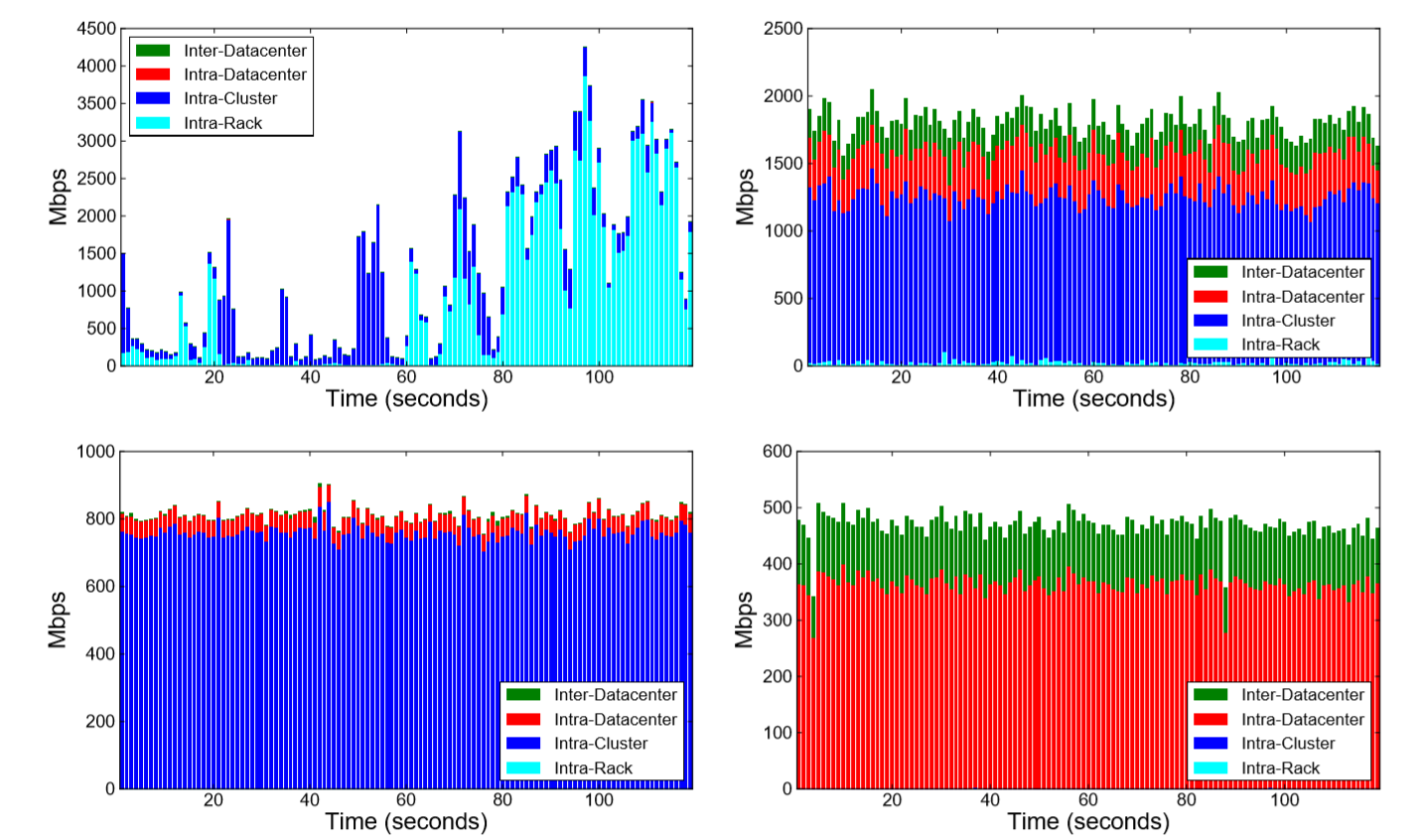


图9 流量传输特征（左上图为Hadoop服务器，右上图为Web服务器，左下为cache follower，右下图为cache leader）

除了Hadoop业务外，其他业务服务器的流量传输过程都较为稳定，并且只有很小一部分流量发送至发送方服务器所在rack内的主机，甚至数据中心网络之间传输的流量都比在该rack内传输的流量要多。

从具体数据来看，Frontend cluster中的web服务器发送的68%的流量都发往了其所在的同一个cluster内部的主机，而这其中的80%的流量都发往了cache服务器，而Multifeed服务器和SLB服务器均各接收了8%的数据。该web服务器发送的剩下的流量大多都是发往了其所在数据中心网络内的其他cluster的主机或者发送给了其他的数据中心网络，在同一个rack内传输的流量微乎其微。

由于cache follower主要是负责响应web服务器的请求，因此其发送的流量多发往了同一个cluster内的web服务器。具体数据来看，cache follower发送数据的88%都是发往了web服务器，而且其发送的流量的接收方覆盖了所在cluster内的75%的主机，其中也覆盖了90%的web服务器。

Cache leader负责维持cluster之间信息的一致性，并维护后台数据库，这就使得cache leader的流量传输多分布在数据中心内的不同cluster之间或者是不同数据中心之间，而具体的数据统计也证实了这一点。

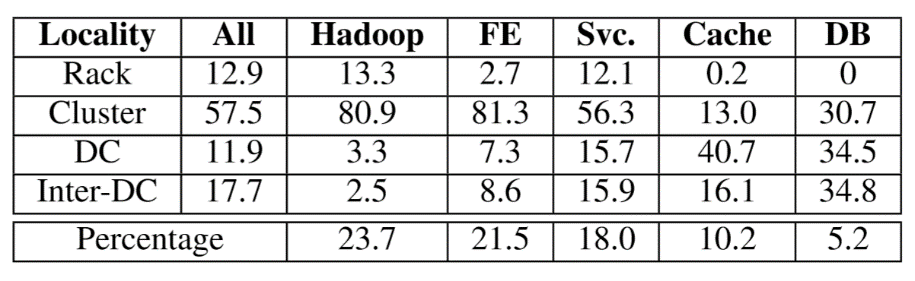
图10列出了2015年1月在Facebook数据中心采集到的24小时间隔内流量传输分布比例，同时也分别列出了不同的业务的占比情况。

图10 不同业务的流量分布比例

很明显，大部分的流量都发往了发送方所在的cluster内的其他rack下的主机，约占总流量的57.5%，而在同一个rack内传输的流量占比约为12.9%，小于数据中心网络之间传输的流量比例。这与之前研究所说的流量主要在rack内传输的结论不同，即使是rack内传输的流量比重最大的Hadoop业务也只有13.3%的流量发送给了发送方所在rack内的主机。

通过图10也可以发现流量比例最高的五类业务分别为Hadoop业务（23.7%）、Frontend业务（21.5%）、Service业务（18.0%）、Cache业务（10.2%）、Database业务（5.2%），这五类业务流量约占总流量的78.6%。

1.2.5. 流量矩阵特征

图11左侧两幅图分别为Hadoop业务和Frontend业务具有64个rack的cluster内的流量矩阵，其描绘的是cluster内rack之间传输的流量大小。

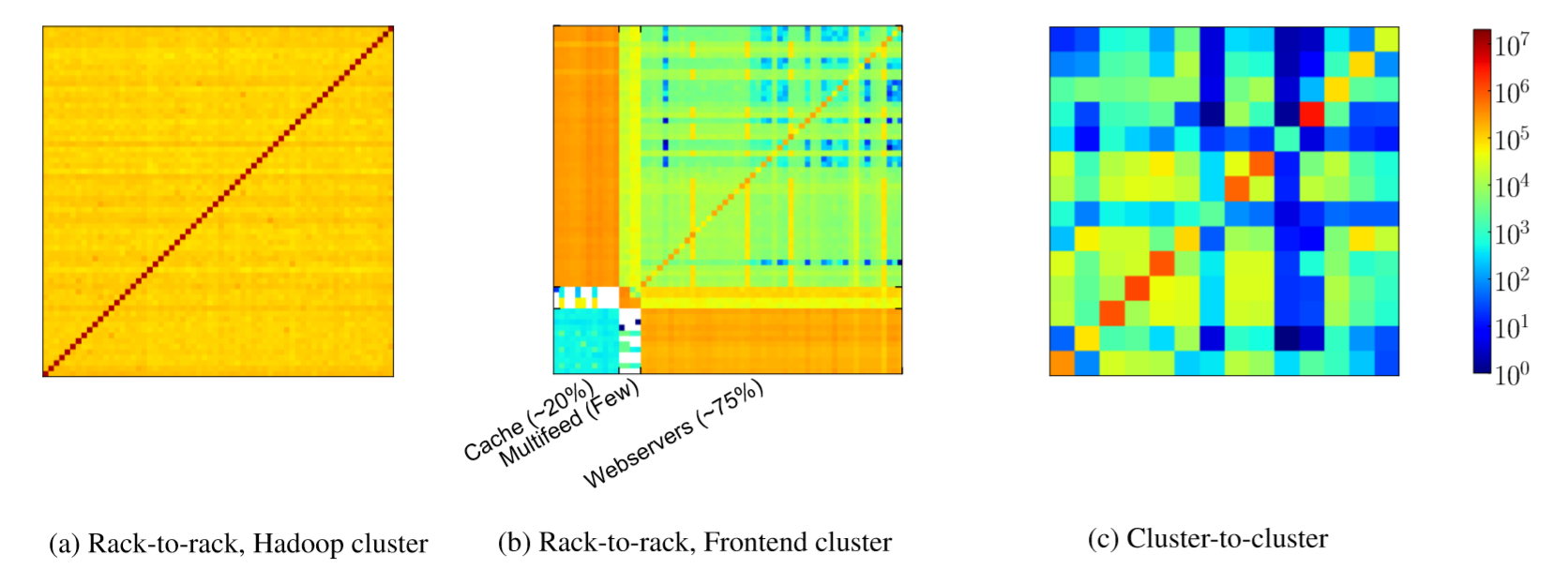


图11 流量矩阵特征

由左侧的Hadoop业务流量矩阵可知，Hadoop cluster内的不同rack之间的传输流量分布非常均匀，约占Hadoop cluster内传输的总流量的80%，而对角线上的值相对较大，说明rack内传输的流量相对于发送给其他rack下主机的流量具有更大的比例。这样的结果可能是因为map任务主要分配在本地，但是过量的任务本地rack无法提供足够的资源，因此，必须将一部分的流量发送给其他rack的主机进行处理，还有因为有一些任务并不要求在本地rack内的主机处理，因此这些任务也会使得本地rack内流量传输的比例下降。

图11中间的流量矩阵展示了Frontend cluster的三类不同服务的rack流量传输分布情况。由图数据可知，rack内部传输的流量并没有明显的优势，而web服务和cache follower的rack之间传输的流量占较大的分量，这说明web服务器主要向cache服务器发送流量，反之亦然，而不同类型的服务器存在于不同的rack中，导致了rack内传输的流量较少。

还有三种服务cluster的流量矩阵未在图中展示。Cache leader cluster的流量矩阵反映出在rack内流量传输的需求很小；Backend database的流量传输分布非常均匀，几乎cluster内所有的节点都平均地分配传输流量；Service cluster地不同rack支持的服务都有可能不同，因此它展示的流量模型介于这些极端情况之间。

图11右侧的流量矩阵展示的是在24小时的时间间隔里15个不同cluster之间传输流量的大小。由图可以发现由于cluster承载的业务不同，他们之间的流量大小会有较大差异。因为每个cluster内部都采用的是4-post的clos架构，而不同的cluster对之间传输的流量大小差距可能超过7倍之多，所以考虑在不同的cluster之间采用不同质量的通信链路可能会有一定的意义。

由于边缘接入链路具有很低的链路利用率，因此我们可以据此调整上层汇聚层和核心层链路超额分配的比例，但是这也需要观察边缘链路的利用率会不会随着使用年限的增加而逐渐增加。流量传输分布的差异性使得采用相同拓扑结构的不同cluster可能会出现在某些cluster中链路资源超额分配，在某些cluster中却出现了拥塞，或者两种情况都有，所以采用在特定位置分配更多带宽资源的非均匀fabric拓扑结构可能会提高网络传输性能。网络流量特征在时间上的稳定性也反映出网络状态的快速调整可能并没有太大必要。

1.2.5 单个流的特征

图12和图13分别展示的是10分钟内（web服务是2.5分钟内）采集到的三类业务的flow（根据五元组区分）大小尺寸和持续时间情况统计：web服务的rack、单独的cache follower（cache leader特征相似，因此没有展示）和一个Hadoop节点。

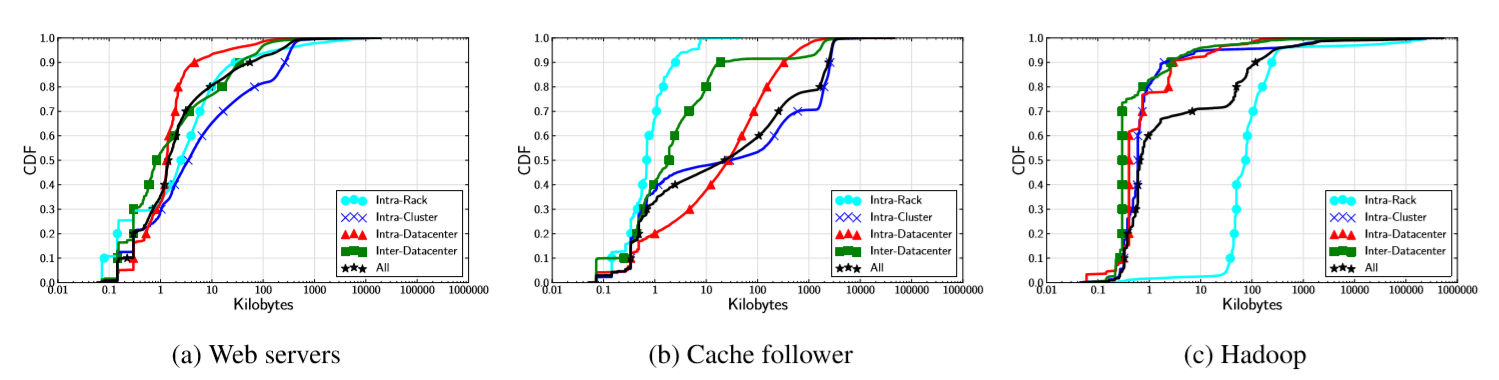


图12 单个流的大小分布

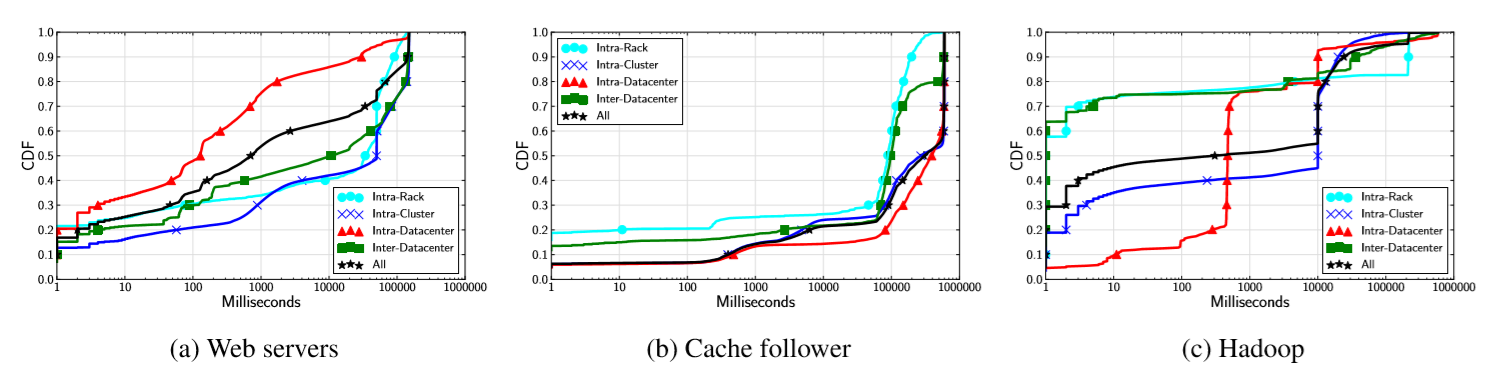


图13 单个流的持续时间分布

由于Hadoop的流量无论是在不同时间间隔上，还是不同的节点上，都会呈现出较大的差异，因此这里只测量了某个Hadoop节点在传输流量较大的10分钟间隔内的特征。从统计数据来看，Hadoop业务的flow约有70%长度尺寸都小于10KB，持续时间小于1s，只有不到5%的flow尺寸大于1MB或者持续时间超过了100秒，几乎没有flow的持续时间超过10分钟。

与Hadoop业务相反，其他服务的流量因为存在负载均衡的影响，要更具有代表性。而且许多Facebook内的服务都采用了一些connection pooling的机制，使得其flow在相对较小的吞吐量下却由较长的持续时间。

Cache follower和cache leader就是采用connection pooling机制的服务，大体上cache的flow尺寸要远大于Hadoop，follower中持续时间小于100秒的flow约占30%，leader中约占40%，而40%的follower的flow持续时间超出了这里10分钟的测量间隔，leader中有25%的flow持续时间超出了10分钟。这也就是说，大多数cache的flow都只在某些毫秒级别长度的时间间隔上较为活跃（在传输数据包），而这些时间间隔之间隔着较长的空窗期。换句话说，无论flow的尺寸大小还是持续时间是多少，flow内部都呈现突发式传输的流量特征。Web服务器的flow特征介于Hadoop和cache之间。

1.2.6 负载均衡特征

流量工程目的是根据网络流量的变化情况调整业务路由，使得网络链路负载均衡。如果业务大小变化（随时间变化或不同区域内流的大小区别）很小，那么流量工程的作用就会减少。都是在利用流量的差异性，因此，过于稳定的流量性能提升的空间就会很局限。先考察2分钟间隔内在目的rack上观测到的每秒的flow到达速率情况，如图14所示。

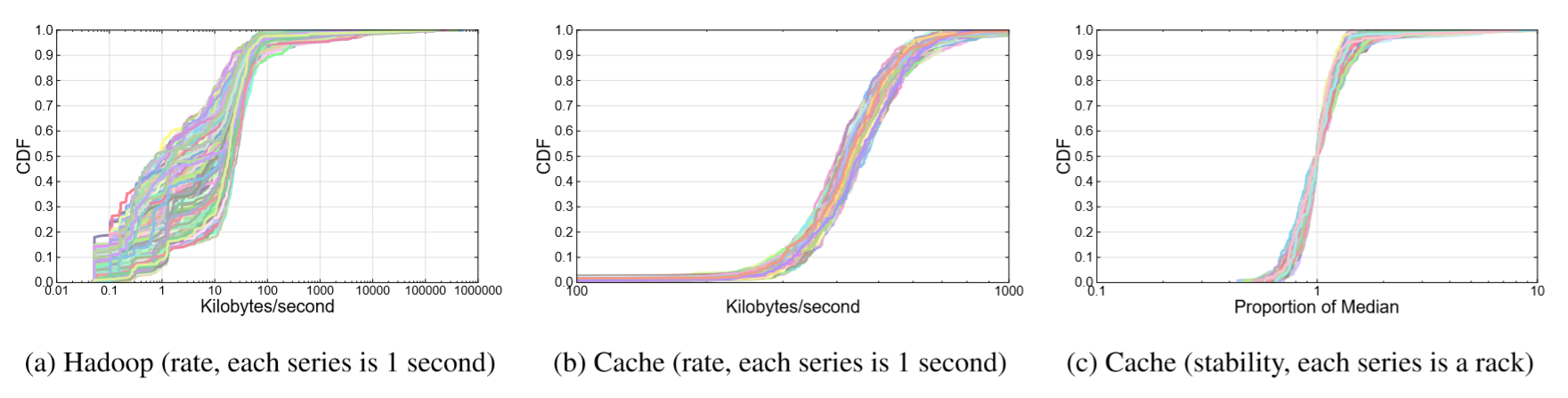


图14 不同业务

图14左侧两幅图分别是每一秒上Hadoop业务和Cache业务每个目的rack的flow到达速率的分布情况，一共检测了2分钟的时间间隔，因此共有120条线，每条线对应其中某一秒的数据。

由图14左侧图可知，Hadoop业务的flow接受速率每秒的变化很大，变化大小超过3倍的情况很普遍。而由中间的图可知，Cache业务在120秒的测量期间内变化较小，曲线相对较为密集，差不多在中位数2Mbps处与横轴垂直。其他的业务与cache业务的曲线相似。

由上述测量可知发往各个目的rack的Cache业务流量在时间上基本保持平稳，而图14右侧图还单个cache业务的目的rack进行了测量，以检验单个目的rack的流量到达速率在时间上的变化大小，其数据按中位数进行了归一化处理。从图中曲线可知，曲线在中位数附近非常密集且接近垂直，所以可知单个目的rack的流量到达速率在时间上也较为平稳。

通过上面的数据可知，cache业务的每个目的rack的flow到达速率非常平稳，以1秒为间隔进行观察，所有的flow在90%的时间间隔内的flow到达速率都在其中位数的2倍以内，而Hadoop业务与之差别很大，最中心的90%的flow的到达速率值的可以超出其中位数的6倍之多。

1.2.7. Heavy hitter流特征

对传输速率最大的flow或flow的集合是关注的重点，因为它们最有可能对网络性能产生影响。定义heavy hitter为固定时间间隔内超过观测流量总大小的50%的flow（或主机，或rack）的最小的集合。因此heavy hitter就是网络测量关注的重点。图15展示了1毫秒时间间隔内四类服务heavy hitter包含流的数量或尺寸。因为瞬间传输的大流对网络影响更大，所以heavy hitter的尺寸就用实际流到达的速率来代替。



图15 Heavy hittr

图15展示了heavy hitter生存时间特征，分别在1毫秒、10毫秒、100毫秒三种测量间隔下，该时刻检测到的所有heavy hitter在下一时刻依然存在的比例的分布情况。

由图可知，Heavy hitter flow的持续性较差，无论检测的时间间隔大小为多少，下一个间隔依然存在的heavy hitter比例中位数不超过15%，这说明内部flow的突发性特征。基于主机的heavy hitter情况会稍好些，除web服务外能持续到下一个时间间隔的heavy hitter host比例中位数不超过20%。Web服务在100毫秒测量间隔有较好的持续性，heavy hitter host持续到下一间隔的比例中位数约为30%。Rack级别的heavy hitter情况最好，100毫秒的检测间隔下，持续到下一间隔的heavy hitter比例的中位数均超过了40%，其中web服务甚至超过了60%，而1毫秒的测量间隔下持续到下一个间隔的heavy hitter比例中位数大约为32%，这说明web服务的heavy hitter大体上要更加的稳定。尽管如此，heavy hitter的持续性依然不足，不是特别支持流量工程。

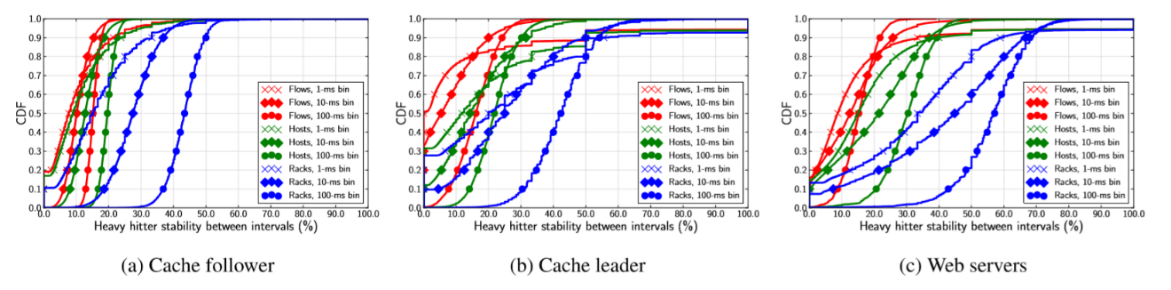


图16 不同业务Heavy hitter稳定性分布

Facebook数据中心广泛应用的connection pooling机制导致flow的持续时间较长，这为流量工程提供便利条件。应用层的负载均衡做到了较好的均衡效果，却限制了一些方法提升网络性能的空间。很多现有的技术是通过检测网络中的heavy hitter然后对他们特殊处理来改进网络性能，这些技术的前提是要先检测出heavy hitter，然而在持续性未知的各类cluster中检测出heavy hitter并不是一件容易的事情，并且即使在像rack级别的heavy hitter并且在大于100毫秒的间隔上检测，也不能确定能带来多大好处，因为heavy hitter在大多数的间隔上并不是始终都能持续。之前确实有工作指出如果35%的heavy hitter都可以持续，那么流量工程就有做的意义，然而，也只有web和cache服务rack级别heavy hitter达到了这个要求。

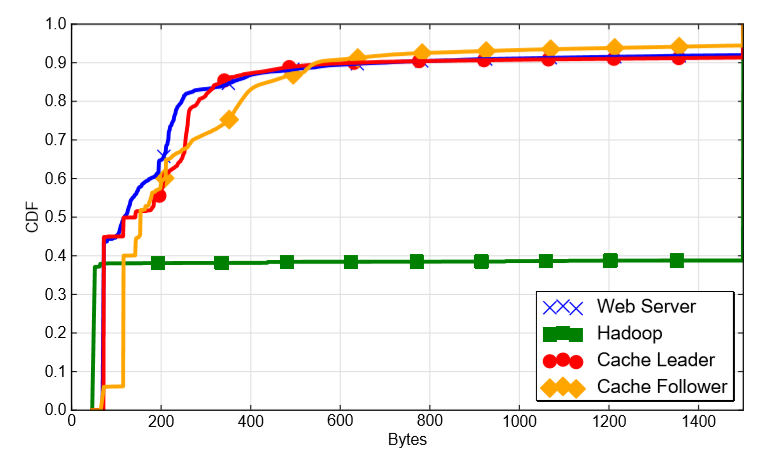


图17 不同业务的数据包大小分布

1.2.8数据包特征

图17描绘的是四种服务主机发送的数据包大小分布情况。总体来看，数据包的平均尺寸在250字节左右，但是这个数字主要是因为Hadoop流量的影响发生了较大的偏移。

Hadoop服务的数据包尺寸呈现双峰分布的特征，即所有的数据包要么为MTU上限（这里是1500字节）的长度，要么为TCP ACK的大小，大概不到50字节。其他服务的书包尺寸分布较为宽广，但它们的中位数均没有超出200字节，即大概只有MTU上限的5%-10%。由此可见，尽管链路利用率很低，但是数据包的发送速率还是很高的。以cache服务器为例，假设服务器发往RSW的链路的利用率为10%，并且数据包平均尺寸为175字节，则该服务器每秒发送的数据包速率为发送MTU尺寸的数据包使链路利用率达到100%的数据包速率的85%。因此，对于单个数据包的操作，通过链路利用率可能无法立即看出发送过程承受的压力大小。

1.2.9 数据包到达速率特征

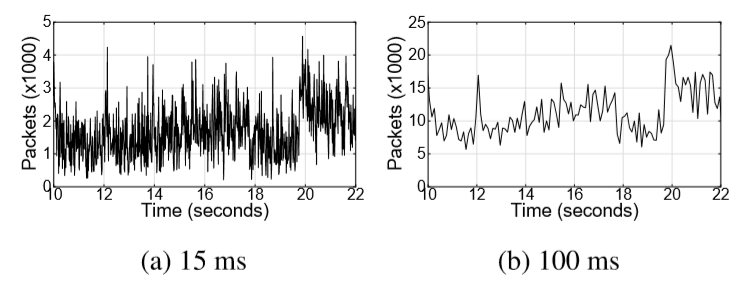


图18 数据包到达速率变化情况

之前的研究结果表明，在主机处观察到的数据包到达率特征呈现开关模型。然而，对Facebook的数据中心的监测发现，即使是Hadoop业务也未能反映出这一特征，图18展示了以15毫秒和100毫秒为间隔统计的一个Hadoop主机发往对应RSW端口的速率随时间变化曲线。而如果把发往不同宿点主机的数据包分离来看，便可观察到开关模型，这说明是发往不同宿点主机的数据包混杂在一起淹没了开关模型的特征。

图19展示的是四类业务的主机检测到的tcp flow之间到达时间间隔的分布情况，即TCP SYN数据包达到速率情况分布。虽然Facebook数据中心有很多服务采用了connection pooling机制，但依然存在一些短暂的flow。通过图可以发现，Hadoop和web服务器flow到达间隔的中位数均在2ms左右，也就是说每秒约有500个flow到达。可能因为存在connection pooling机制的关系，两类cache服务器的flow到达时间间隔均高于web服务和Hadoop服务，而cache leader服务器的到达率又略高于cache follower，cache leader服务器flow到达时间间隔约为3毫秒，cache follower的约为8毫秒。

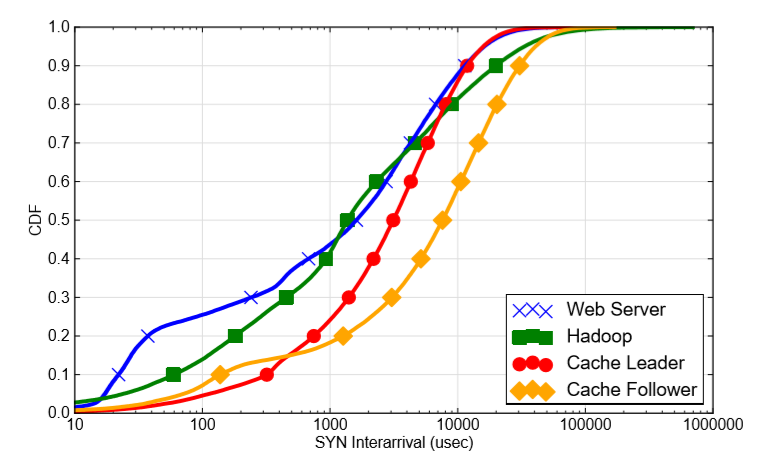


图19 不同业务的TCP流的到达间隔分布

1.2.10交换机Buffer占用率

对Facebook数据中心网络中分别连接web服务和cache服务服务器的交换机buffer占用率在24小时的时间段内以10微秒为时间间隔进行采样，并绘制每秒测得的占用率中位数和最大值，得到图20所示的结果。同时还分别对其在该测量时间段的链路利用率和丢包率情况进行统计，统计结果根据其最大值进行归一化处理，结果如图21和22所示

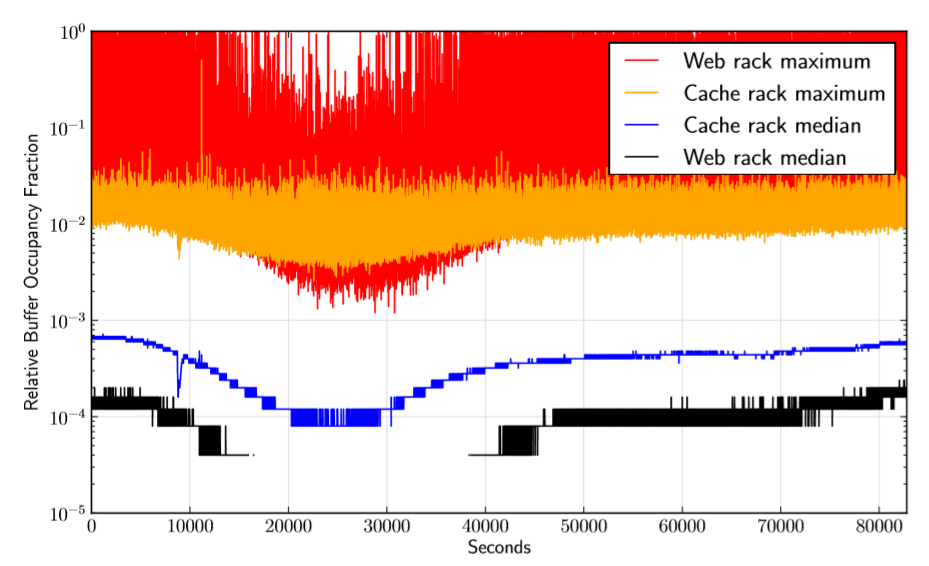


图20 承载不同业务的交换机上buffer的占用比例

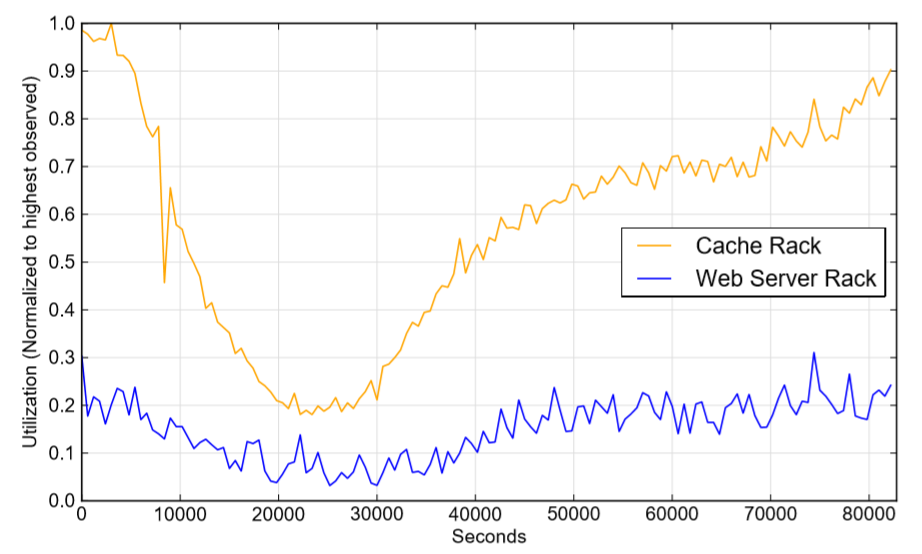


图20 承载不同业务的交换机上的端口利用率

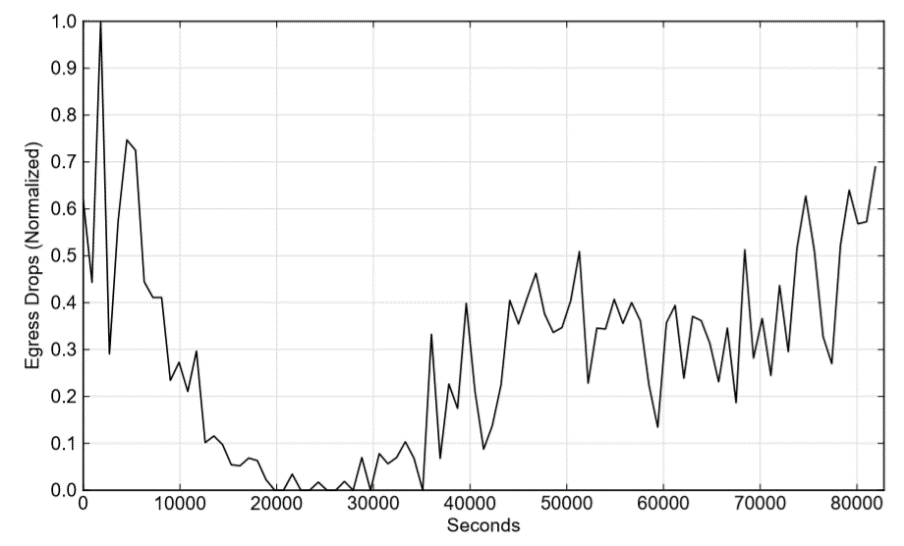


图21 交换机的丢包率

通过以上测量的结果可以发现，buffer占用率在测量的24小时时间段内持续较高，特别是web服务的交换机buffer占用率非常高，尽管其链路利用率大多数时间都在1%附近，但是在每个10微妙的采样间隔内，超过2/3的可用的buffer都被占用，且24小时内有3/4的时间段内buffer占用率最大值都有可能超出buffer占用的上限，而且web服务链路利用率和buffer占用率的变化过程大体一致。不过cache服务的链路利用率和buffer占用率变化关系没有那么明显，而且尽管cache服务的链路利用率要高于web服务，但是其buffer占用率和丢包率要远低于web服务。同时，这两种服务的buffer占用率、链路利用率和丢包率均与时间有关，24小时内不同时刻可能会出现较大差别。

持续较高的buffer占用率可能由于数据包到达率非开关模型、flow到达率较高、单个flow呈突发式传输等因素引起，因此如果数据包尺寸增大、flow的到达速率增加或者多个flow同时突发性传输流量都有可能对buffer占用率造成较大影响。

1.2.11. Flow的并发性

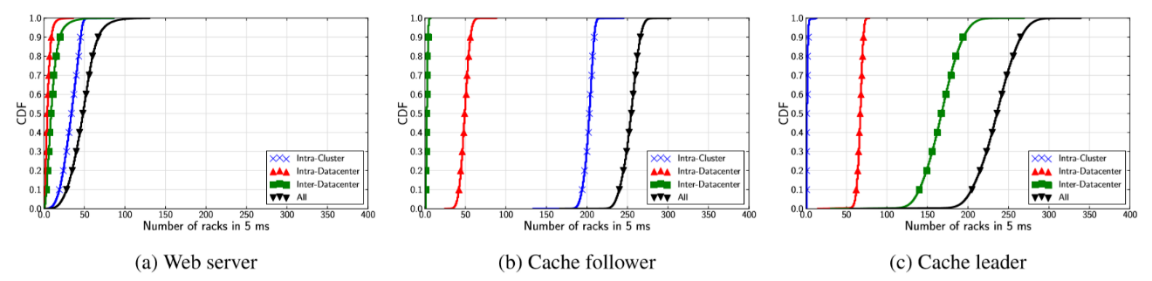


图22 不同业务流的并发特征

这里的并发性是指在5毫秒的间隔内同时传输的flow。因为之前已经验证rack内传输的流量很少，因此可以分别检测三类服务的单台主机在每5毫秒的时间间隔内与其传输数据的不同位置的rack的数量，绘制图22所示的rack数量统计图。

由图22中数据可知，与cache follower主机在5毫秒时间间隔内同时传输数据的rack数量在225-300之间，而与cache leader主机同时通信的rack数量在175-350之间，而他们的rack数量中位数都在250左右，但是他们rack分布的位置却有较大差别，前文测量结果也已验证过这一点。Web服务器同时通信的rack数量在10-125之间，中位数为50。

由于有一些交换机为大流有专门的设计，所以现单独考虑占大部分传输流量的heavy hitter rack的数量，测量结果如图23所示。

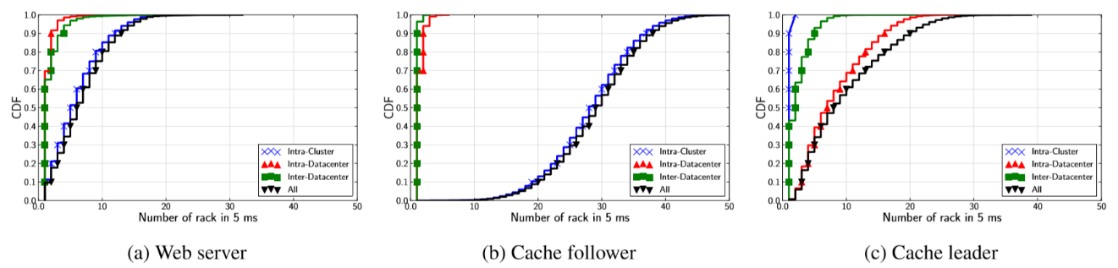


图23 Heavy hitter rack的数目分布情况

对于web服务，heavy hitter rack的数量在6-8之间，而cache leader的heavy hitter rack数量最大值在20-30之间，cache follower的heavy hitter rack数量中位数在29左右，最大值可以接近50上下。再看heavy hitter rack的分布，cache follower和web服务器在与其同一个cluster内的heavy hitter rack占大多数，而cache leader与之相反，cache leader服务器的同一数据中心网络内的heavy hitter rack占大多数。

## 手机云存储业务

手机云存储不同于传统的桌面云存储业务，它有许多新的特点：

1. 手机云存储的传输流量特征遵循包含两个component的混合高斯模型；
2. 上传至云端的文件数目要远高于下载的文件数；

本实验的数据采集自日活跃用户超过百万的一款中国的云存储服务商，用户可以通过其电脑或手机进行上传、下载、删除和分享操作，并且支持多文件的批量操作，上传文件时本地文件也不会丢失。其中数据传输使用的是HTTP服务，每个HTTP的报文都是固定大小512KB（除了每次HTTP传输的最后一个报文），如果超出了固定大小限制的文件则需要被拆分成多个符合大小要求的报文进行传输。对于一个文件，HTTP请求可以使用多个TCP连接进行传送，同时，一个TCP连接也可以传送多个文件的HTTP请求，这时需要按一定的序列逐个发送，并且只有在收到上一个报文的应答后下一个报文才会发送。

无论是上传和下载，用户设备首先会与metadata server联系。上传时，用户先发送一个包含文件名和文件MD5值的metadata至metadata server，metadata server先检查该文件是否已存在于云存储中，如果已存在，则直接将文件加入到该用户的云存储空间并通知用户设备无需上传，如果不存在，则发送给用户设备离其最近的front-end server的身份信息，通知用户设备与之建立连接。用户设备收到front-end server的身份信息后便向其发送一个file storage operation request，里面包含了文件名、文件大小、文件MD5值、待传的chunk数量、每一个单独的chunk对应的MD5值等信息，然后用chunk storage request初始化存储过程。下载时，用户设备先请求预设的metadata server下载文件对应的MD5值，然后使用这个值通过file retrieval operation request去从storage server请求关于该文件的信息，之后再通过chunk retrieval request请求一个一个的数据报文。

这里定义从一个file operation request的开始到下一次file operation request的开始的区间算作一个file operating interval，用T表示；给定时间参数τ，从某个的file operating interval的结束开始到下一个的file operating interval的最后一个报文的末尾作为结束，称其为一个session。

本实验采集的数据是2015年8月里该云存储业务的所有移动端用户的一周的HTTP请求日志，一共从1148640个活跃用户使用的1396494个移动设备中获得了349092451条日志记录，其中一个用户可能在使用多个移动设备。

### 2.1 流量大小和文件传输数量的时间分布情况

首先将得到的数据绘制成数据量随时间变化的曲线，如Figure 7所示：

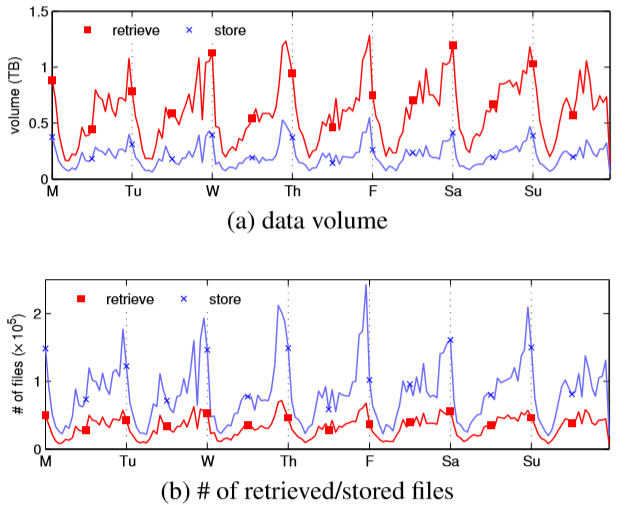


Figure 7

由图可以发现用户的数据流量大小和文件数目整体随时间呈现大致的周期性变化，在每天的23:00左右呈现出一个较高的尖峰，随后在3:00左右为一个低谷，这与手机用户的日常使用习惯相符。其次，对于流量大小，用户下载的流量大小始终都明显高于上传的大小，而对于文件数目，上传的文件数却又明显高于下载的文件数，由此可以推断，手机用户喜爱上传大量容量较小的文件，而下载时都偏向下载大容量文件，且上传的文件数目远高于下载的文件数。

根据分析结果可知，无论是storage server还是metadata server，为了能够应付数据传输的峰值情况，在大多数时候的实际资源利用率都很低，这时就需要使用可以弹性调整其资源大小的服务来减少这种浪费，其次，大容量数据传输对该云存储存储空间和带宽需求也是一个巨大的挑战。

### 2.2 File storage/retrieval operating interval的大小分布情况

通过采集到的数据机绘制出file operating interval的大小分布情况，如Figure 8所示：

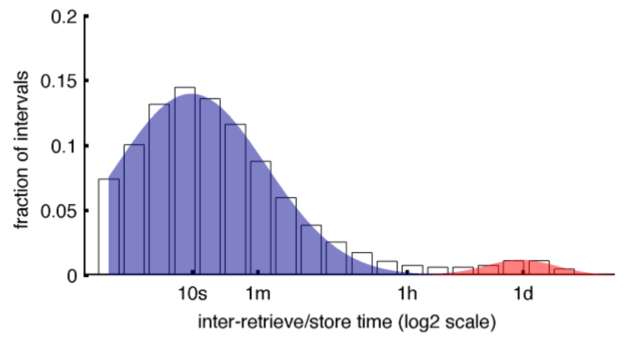


Figure 8

由图可见，file operating intervals的分布是一条双峰曲线，其中一个波峰对应的interval大小为10s，另一个波峰对应的interval大小为1天，而中间频率几乎为0的波谷对应的interval大小为1h，由之前关于session的定义，这里就推荐令，这样，当file operating interval小于1h时，它与它之后紧邻的下一个file operating interval就同处在一个session内，否则，它传输的最后一个报文的结束就标志着其所在session的结束。具体关系见Figure 9：

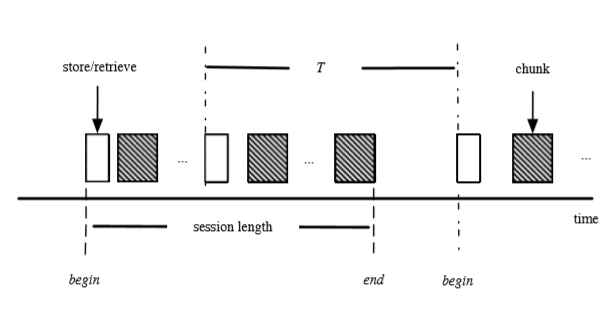


Figure 9

如此一来，手机传输的流量变化特征就服从包含两个component的混合高斯模型，可以通过expectation maximization(EM)算法求得其各个参数，其中一个component相当于横坐标左边的部分，其均值约为10s，另一个compenent相当于右侧的部分，其均值约为1天。

### 2.3 Session的大小变化规律

Figure 10中图(a)绘制的是单个session所包含的file operating interval的数量的CDF曲线，图(b)和(c)描述的是上传和下载两种情况下session的数据量大小随着其包含的file operating interval数量的变化情况：

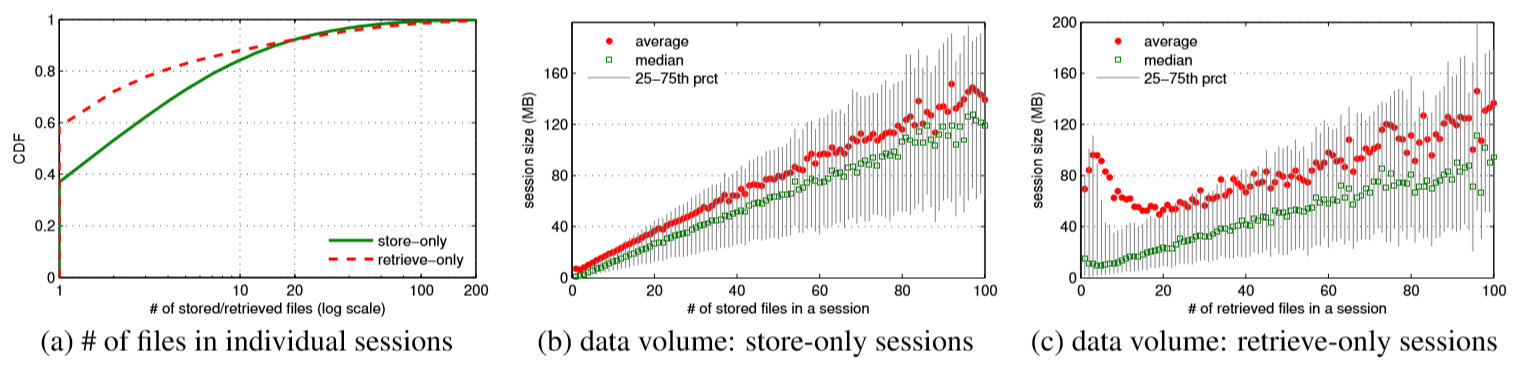


Figure 10

由图(a)可以观察到，差不多半数的session都只包含单个file operating interval，还有大约10%的session包含20个以上的interval，并且下载过程中的session包含的file operating interval的数量总体上要小于上传时的Interval数量。

根据图(b)可以发现，在上传过程中单个session的数据量大小和其所包含的file operating interval的数量呈线性关系，可以推断上传的每个文件数据量大小基本一致。而在下载过程中情况有所不同，单个session包含少量file operating interval时，其中的文件大小平均值较大，而且随着包含的interval数量的增多，session数据量大小波动也非常大，说明下载过程中传输的文件的大小不尽相同。

# 基于光交换的新型数据中心网络架构

近些年，光交换技术在数据中心网络中的应用越来越受到学术界的关注，并取得了一定的研究成果：光网路可以在很大范围的多层、多个域网络内提供动态路由；光网络的连接很有希望实现数据中心之间高触发、高带宽的连接需求；光网络可以通过网络虚拟化来提高网络设备的效率等等。有一些数据中心已经用到了光交换技术，并且都依赖于一个中心的控制器，这里我们研究的也是基于一个中心控制器的基于光交换的数据中心网络。

## 网络架构简介

我们介绍的这个架构是基于光突发技术（Optical Burst Switching）。首先，光突发交换技术需要将网络分为两层——控制平面和数据平面，数据平面拓扑又包括了服务器机架（rack）上的Tor-switch和快速光交换机（Fast Optical Switch），分别属于edge层和core层。常见的快速光交换机有arrayed waveguide grating routers（AWGRs）、semiconductor optical amplifiers（SOAs）和1×N photonic switches，他们的转发数据包的处理时间都在纳秒（ns）量级，可以实现非常快速的转发。

每一个服务器机架中的单个服务器都与其顶层的Tor-switch通过电路连接，而Tor-switch又与其之外的快速光交换机连接，并通过management network与控制平面的控制器连接，控制器可以分别传送控制信息给网络中的Tor-switch和快速光交换机，其具体结构如Figure 11中图(a)所示：

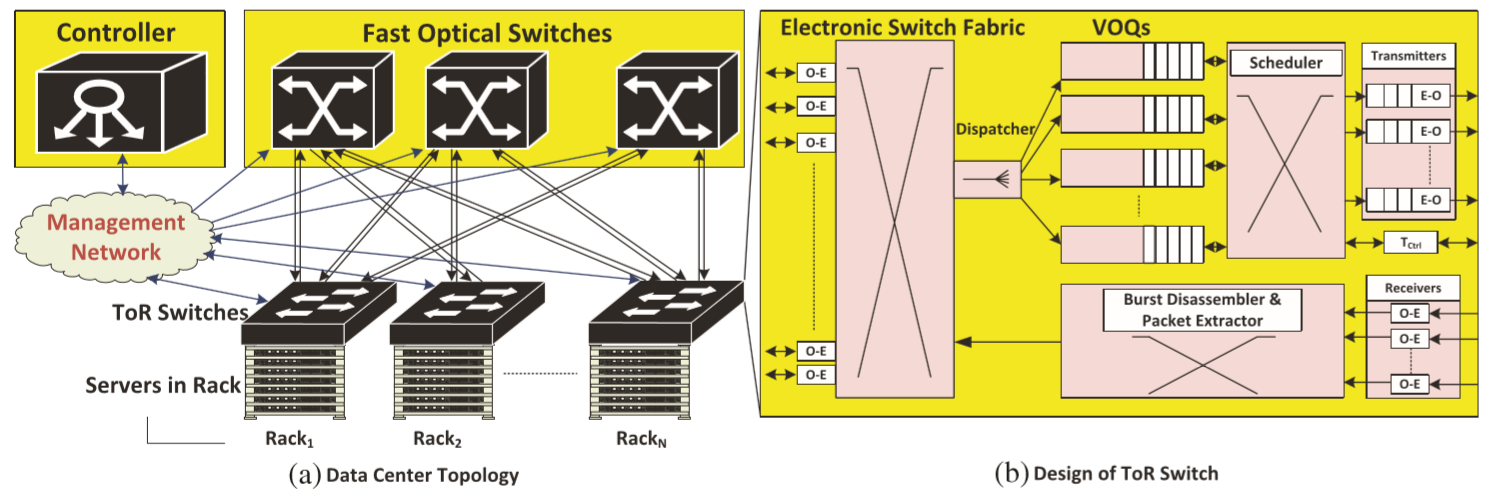


Figure 11

这里推荐使用的快速光交换机是带有1024个接口的SOAs，但是也可以使用带有512个接口的AWGRs。接口数量越多，其连接的Tor-switch数量也就相应的越多，因此负责的服务器数量也就越多，但是，这样的扩展性也取决于控制平面控制器的性能。

Tor-switch内部结构如Figure 11图(b)所示，每一个Tor-switch都通过电路交换网与rack下层的服务器相连接，用以负责rack内的数据交换。同时，Tor-switch也负责rack之间的数据交换，这里采用的是在Tor-switch内设置个虚拟的输出队列（Virtual Output Queues），N是数据中心网络中所有Tor-switch的数量，每一个VOQ都对应一个特定的目的Tor-switch，用于收集想要发送到目的Tor-switch的所有数据。每一个Tor-switch都维护着一个VOQ表来记录每个VOQ对应的VOQ编号和目的rack网络地址。Tor-switch中的dispatcher模块根据VOQ表将从rack下层服务器收到的数据包匹配至对应的目的网络地址的VOQ中。Scheduler则负责根据控制器发挥的控制信息来将数据转移至对应的发送接口发出，其中，Tor-switch还通过一个transceiver与控制器通信。

控制平面主要由中心控制器组成，负责路由、调度（scheduling）和交换处理功能。它接收来自数据中心网络中所有Tor-switch的连接建立请求，寻找对应的俗陋方案、给每个连接请求分配时隙并根据时隙分配调度网络资源实现光交换。控制层维护着一个连接状态记录数据，记录着所有光交换机在对应时隙上的状态，数据平面的光交换机就是根据控制器预先分配的光路来进行数据包的转发。控制平面与数据平面通过一个管理网络（Management Network）相连接。

## 数据转发过程

### burst的产生与发送

burst的形成可以是基于时间的，也可以是基于长度的。当数据包到达一个空的VOQ中时，计时器启动，所有到达该VOQ的数据包开始聚集，当计时器超时或VOQ中聚集的数据长度超过了阈值时，就产生一个控制数据包（control packet），使用transceiver发送给控制平面的控制器。这个控制数据包包含了数据的长度、源和目的IP地址和源和目的Tor-switch ID值等信息，具体结构见Figure 12。在数据中心网络中每一个Tor-switch都被指定了一个ID编号，从0至，N是网络中rack的数量，这些ID编号被控制器用于做路由和调度算法。控制器处理了控制数据包后，分配一个开始时间（start time）和Tor-switch接口号，数据之后会从接口号对应的Tor-switch接口发出，并将这些信息加入到控制数据包中发回给对应的源Tor-switch。当源Tor-switch接收到返回回来的控制数据包后，就根据其中的信息抽取VOQ中的数据产生一个burst，并在指定的时隙从对应的Tor-switch接口发出。尽管VOQ非空，scheduler模块依然会在控制数据包发出后初始化一个新的计时器，因为在控制数据包发出的RTT时间间隔内会有新的数据包达到该VOQ。

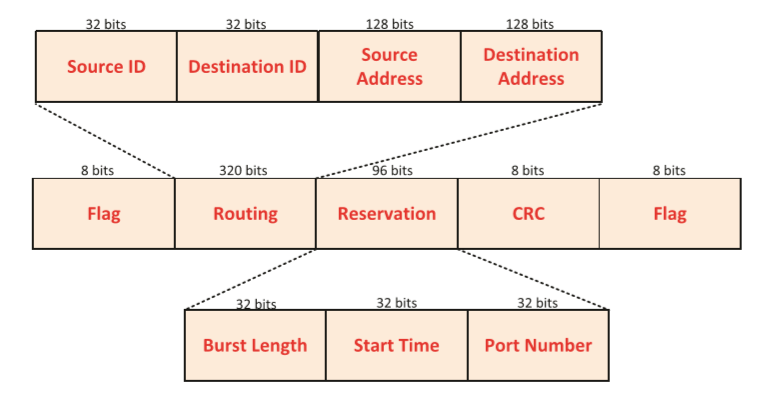


Figure 12

### burst接收过程

当burst到达目的Tor-switch后，Tor-switch中的burst disassembler和packet extractor模块便会将数据包从burst中抽取出来并通过电路交换发送给其所在rack下的各个服务器。

### 控制平面的处理过程

控制器维护着所有光交换机的连接情况信息，并实现路由、调度和操作交换机的处理。其资源分配机制用horizon scheduling来表示，horizon指的是最近的可以使用的空余传输资源的时隙，如Figure 13所示。其中图(a)展示的是在控制数据包到达前的各传输通路的状态，horizon scheduling寻找其中有可用通路资源的最近时间来安排本次burst的发送。图(b)展示了分配时隙之后的状态。

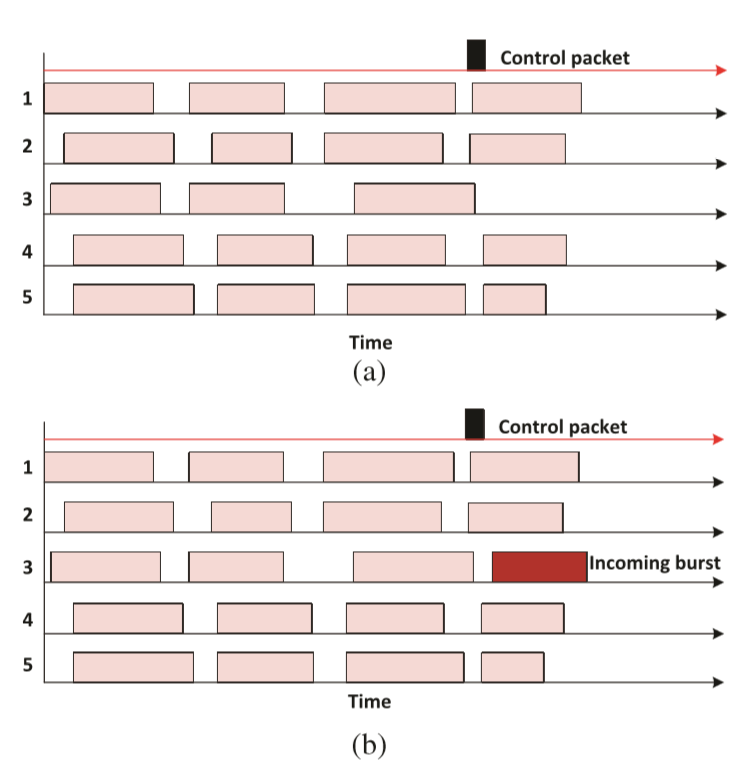


Figure 13

在控制数据包发送后，控制器会产生一个处理信息（configuration message），控制器在其中设置了输入接口、输出接口和执行时间等信息，同时也将其源IP设置成控制器的IP并把目的IP设置为光交换机的IP地址，最后将该处理信息发送至对应的光交换机，光交换机的控制器会根据处理信息中的说明进行操作。

## 性能评估

传统的光突发交换技术相比于这里的改进架构，缺少了对控制数据包control packet的回传，于是burst的产生和发送时间需要Tor-switch设置一个时间偏置来决定，及在Tor-switch将control packet发出后便开始计时，达到时间偏置后就发送burst，控制器则根据收到的control packet直接为其分配网络资源。这样的发送方式导致的结果就是可能有多个burst争抢同样的网络资源导致冲突，从而造成较高的丢包率。而这里的改进算法资源分配是由控制器统一分配管理的，在一定的负载下可以实现0冲突。

这里就对传统的光突发交换的数据中心网络和改进架构的数据中心网络进行对比，来衡量他们的时延、吞吐量和丢包率，同时也与一个典型leaf-spine架构的两层电交换数据中心网络进行了比较，点交换网络如Figure 14所示。

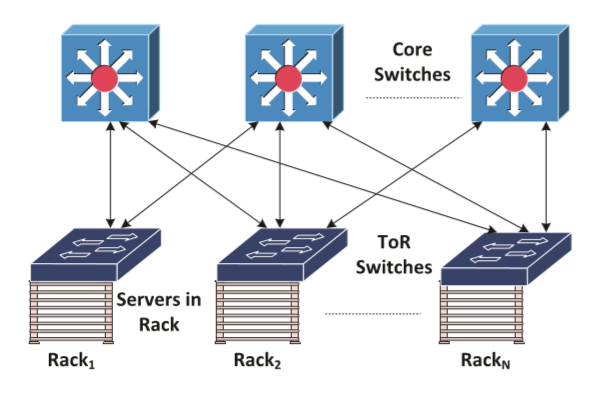


Figure 14

为了测试的结果尽量具备广泛性，该实验测量了10Gbps和40Gbps两种数据产生速率，包含了Tor-switch同时向1台、10台、20台Tor-switch发送数据包的情况，即TDC={1，10，20}，并且还对比了oversubscription ratios为1:1和2:1的情况，即源rack需要发送的数据的目的地址全在其他rack上和50%在其他rack上还有50%在源rack上两种情况。测得的数据如下。

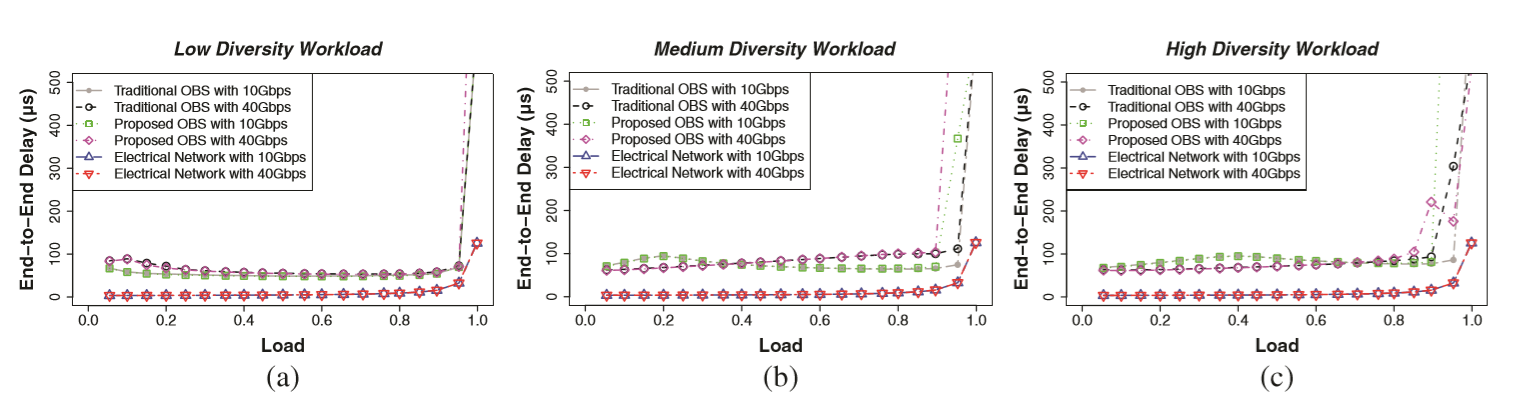


Figure 15

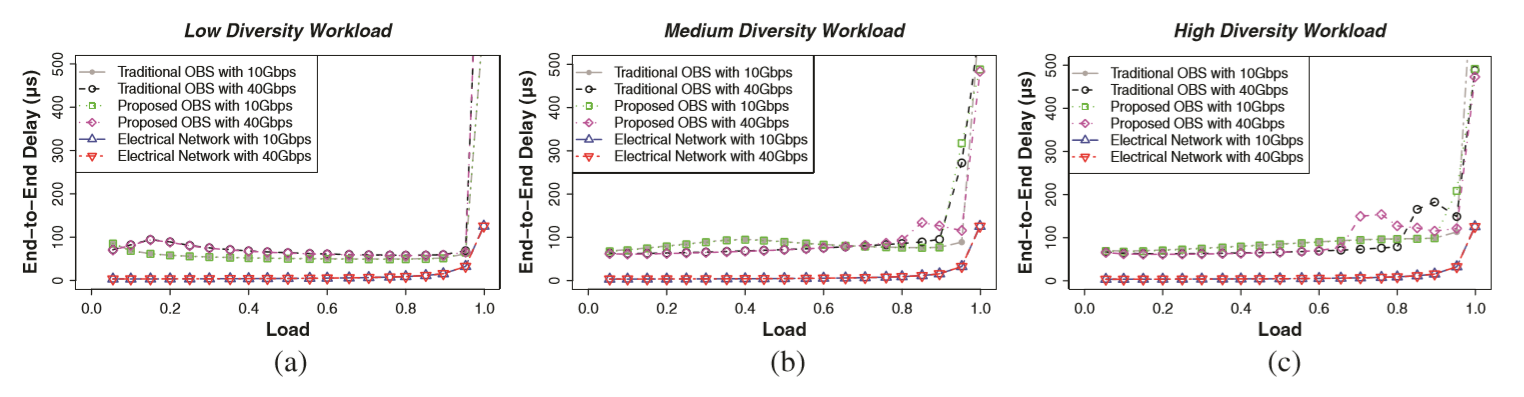


Figure 16

Figure 15和Figure 16分别是oversubscription ratios为1:1和2:1两种情况下的端到端时延情况对比，他们的图(a)、(b)、(c)对应的分别是TDC={1，10，20}三种情况，由对比结果可以发现改进前后时延差别不大，且都比电交换的要差一些，这说明在数据中心这样集中的网络环境下，增加回传机制导致的control packet的RRT延时对总时延影响并不大。

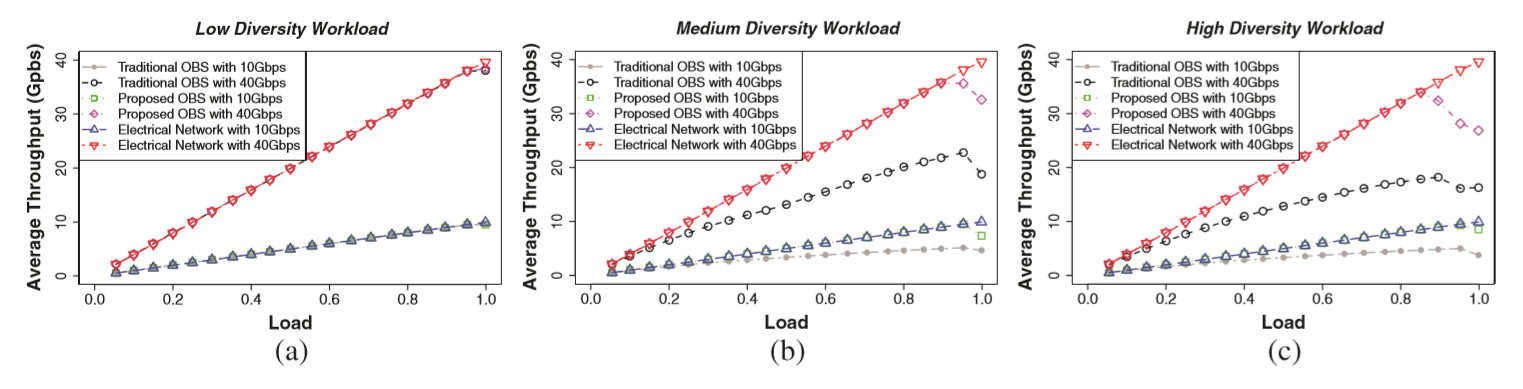


Figure 17

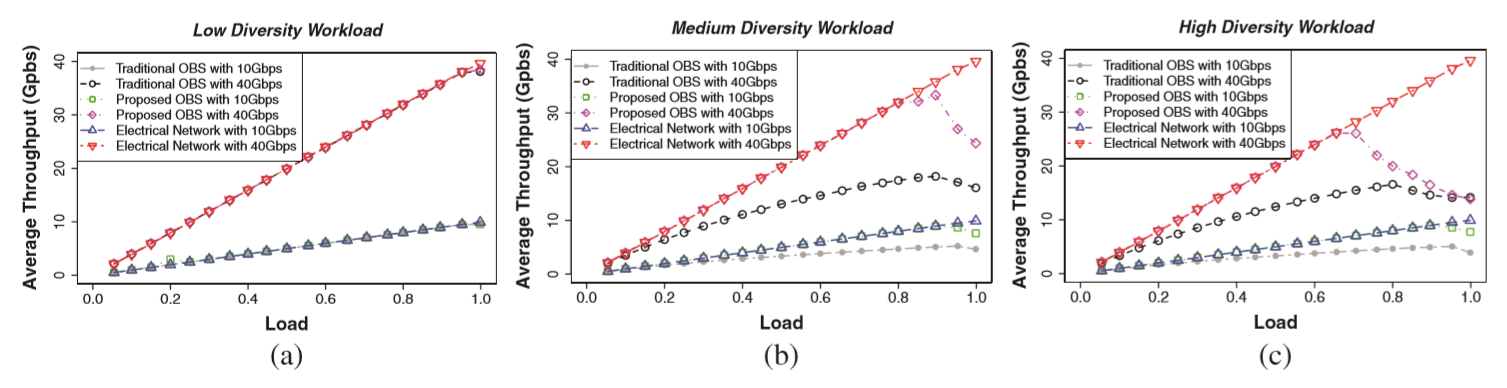


Figure 18

Figure 17和Figure 18分别描述的是oversubscription ratios为1:1和2:1两种情况下的平均吞吐量情况，他们的图(a)、(b)、(c)对应的分别是TDC={1，10，20}三种情况。对比发现改进后的基于光突发交换的DCN架构的吞吐量要比改进前的大，但是改进后的架构在高负载情况下会小于电交换数据中心网络架构。

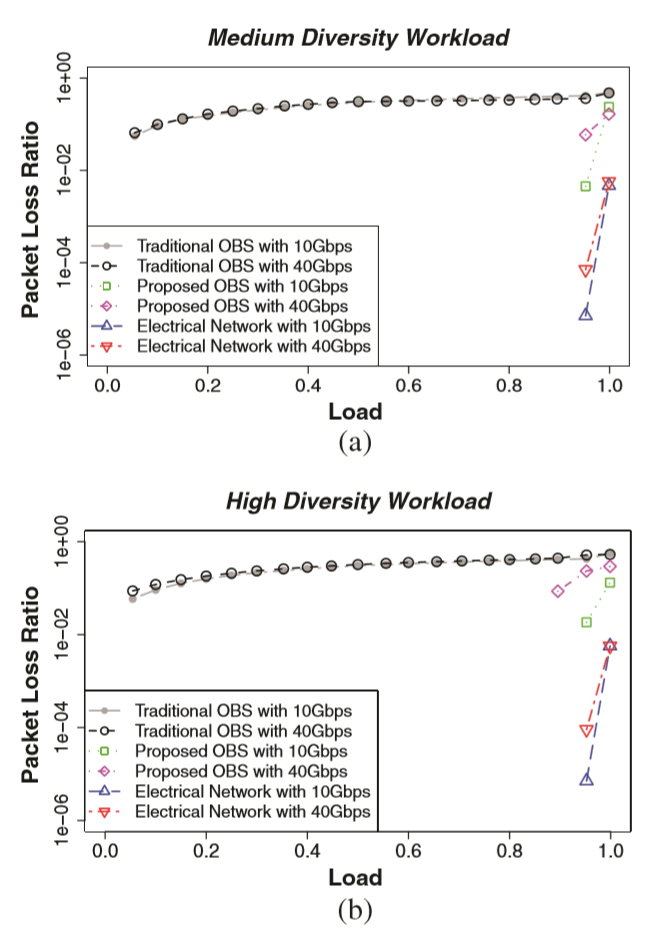


Figure 19

Figure 19中展示了oversubscription ratios为1:1情况下丢包率的大小对比，图(a)对应着TDC为10的情况，图(b)对应TDC为20的情况。由图可知改进的架构和电交换网络的丢包率只有在很高的负载下才会变得较高，而传统的基于光突发的数据中心网络始终保持着较高的丢包率。

由上面的仿真测试数据可知，这里提到的改进的基于光突发的数据中心网络在时延基本不变的情况下提高了吞吐量并降低了丢包率，并且与电交换数据中心网络的差距较小。

## 使用机器学习对光网络的改进

除了以上介绍的基于光突发交换技术的数据中心网络外，还有使用光分组交换的数据中心网络，相比于光突发的传输单元burst，光分组传输的是光分组，并且整体架构依然是由控制平面的控制器和数据平面的Tor-switch和光交换机组成。

在数据包的处理和转发过程上，电数据包先进入到Tor-switch的与目的rack对应的VOQ中等待被转为光分组发送出去，而每一个Tor-switch在一个时隙上只能转换一个VOQ的光分组并发送，因此需要控制平面的控制器进行决策并给予某个VOQ发送权限。当VOQ收到发送权限后就会在下一个时隙发送一个光分组，光分组经过光交换机到达目的Tor-switch。

决定下一个时隙发送哪个VOQ的光分组一般是通过轮询调度（Round-Robin Scheduling）来决定，这既容易实现，保证了处理速度，同时也绝对的公正，每个待发送数据的VOQ机会均等。

但是，轮询调度的方式却忽视了数据中心网络中的数据特征。

根据之前的研究可以发现，数据中发送的流的大小不是均匀分布的，而是呈现出两极化的特点，如Figure 20的图(a)和图(b)所示，数据中心的流量主要由大流和小流组成，且大流和小流的数据长度、持续时间、时延敏感度都有很大差异，因此采用完全公正的方式来决定发送顺序在很多情况下都会不够合理，例如发送了持续时间较长的大流，而使得对时延敏感的小流一直等待。

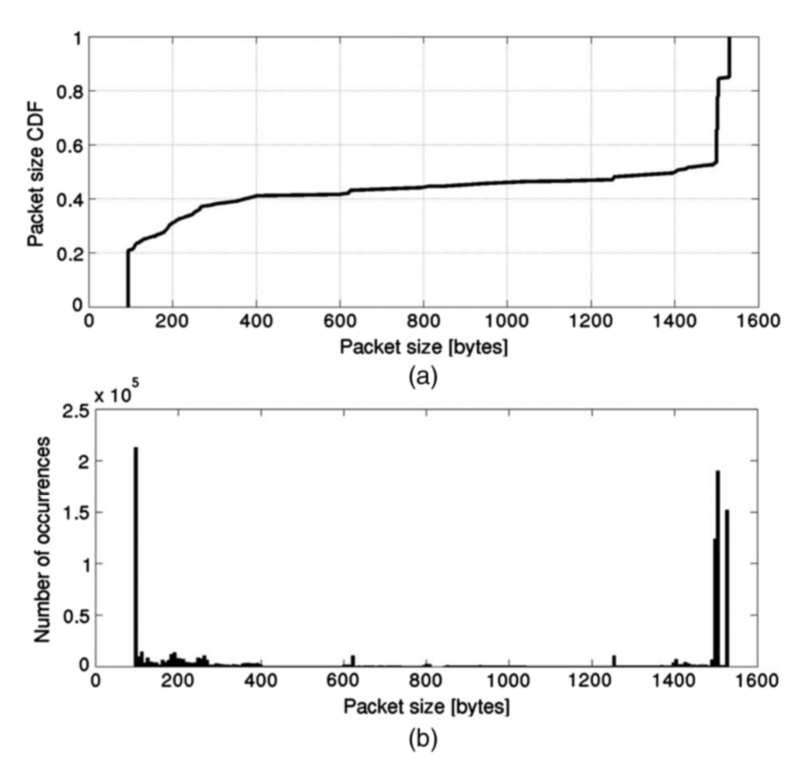


Figure 20

因此可以通过识别VOQ中的大流和小流等特征来赋予每个VOQ一个优先级大小，每次决策都发送优先级最大的。这里，改进的方案考虑优先级的因素有四点：VOQ队列越长优先级越大、数据包数目越多优先级越大、等待时间越长的数据优先级越大、转发过程不需要调整光交换机的优先级越大。通过四个参数（weight factor）将这些指标量化，分别是、、、。优先级计算公式为。

这时调度方案就是根据优先级调度（Priority-Aware Scheduling）。算法具体过程如Figure21所示。

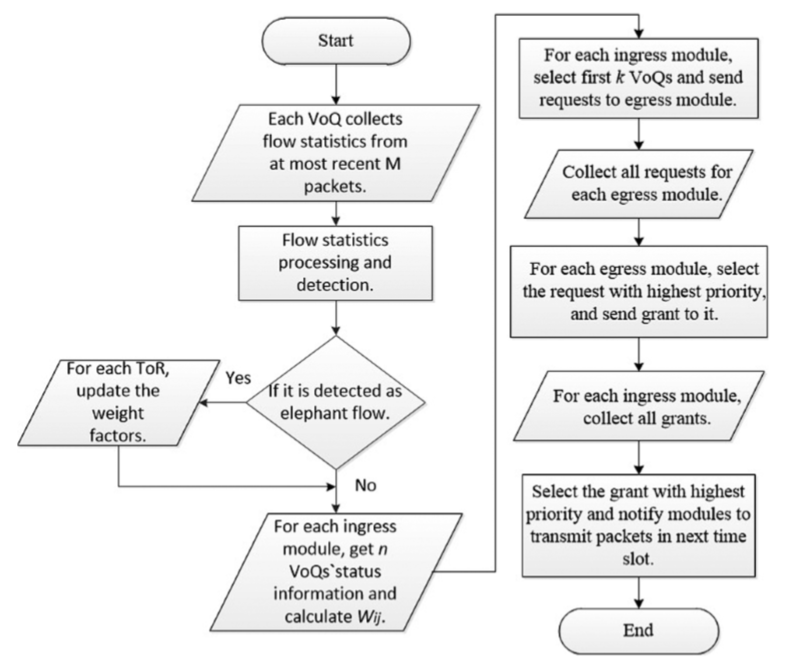


Figure 21

其中，算法中判断最大流的方式使用机器学习的分类算法，这里分别使用C4.5决策树和Naïve-Bayes discretization（NBD），当分类结果发现是大流时，则升高并降低，若分类检查出是小流时则降低并升高，最后计算各个VOQ的权重决定发送顺序。