AMPS：在SDN中使用机器学习的应用感知多路径流量路由

论文后期整理：

摘要：

每一种应用的带宽和延迟需求是不一样的，网络中的流必须要竞争有限的带宽和低延迟路径，一个智能的流路由算法成为了一个很自然的需求。如果网络有能力基于不同应用的特定需求根据流的优先级进行资源的分配，可以具有更好的全网性能。

我们所提出的，AMPS控制器可以使用机器学习技术对每个流进行优先级的划分，然后基于优先级的分类分配路径。AMPS控制器支持在不同的路径上面进行流的路由，即使这些流的源目的地址都一样。

路径发现算法采用的是Yen-K-shortest路径算法，此算法还扩展支持大容量流。

这篇论文提出的算法的目的是在网络负载较重的情况下，确保低负载路径对高优先级流的高可用性。

简介：

不同的网络应用和服务对不同指标(带宽，抖动率，时延，优先级)的要求差异很大。正好，不同路径上的这些指标是不一样的，所以，可以针对不同流的需求给他们分配不同的路径。

如果一个网络拥有足够的对于端到端路径的知识，那么这样的控制器就可以针对每条流的需求安排合适的路径。目前最主要的挑战是：

1. 应用的意识不容易实现，因为很多应用都已经被归入了主要的分类，例如HTTP和HTTPS，把他们更加细分为声音，视频等等是一个很乏味的事情
2. 网络的状态是动态变化的，静态的路径分配可能不会起作用

现在我们可以使用SDN来根据不同的需求去智能的选择路径，让应用的性能达到最优

SDN控制器可以为相同源目的节点的多条流分配相同的路径，即使源目的节点之间存在多条路径。如果源目的节点之间既有应用意识也有多条流路由，那么这里就会有两个特征集可以用于分类不同的应用，也可以被用于根据应用的维度安排不同的路径给不同的应用。

这篇论文提出了一个方法，用于基于应用的意识和路径的状态对多条流进行路由。这个系统采用机器学习方法去评估流的参数，基于流的44个特征中的四个。

现在控制器基于参数来评估每条可能路径，然后根据流的优先级把路径分配给这些流。

这篇论文的贡献在于采用在SDN中采用机器学习方法实现了应用的意识。并且整合了多个分类属性供SDN控制器进行决策。我们提出了一个高效的多路径路由算法去支持大量的流。

相关工作：

Weiyang提出了一个策略来克服流量拥塞和物理损坏。通过查找一个替代路径来阻止拥塞。

Marc Koerner采用了一种数据中心网络架构（胖树拓扑）提出了同一对交换机之间的多路径路由，但是他们的多路径路由是随机的并且没有考虑QoS。

还有人提出了policycop框架，这个框架采用了服务水平协议使用了QoS在SDN环境中。Policycop将会采取决策去避免网络中像丢包，抖动，时延和带宽这样的问题。FlowQoS执行了应用的身份鉴别和流速率。作者开发了一个分类模块，通过DNS信息分类了HTTP和HTTPS。

Wang提出了一个QoS意识的流量分类框架采用半监督机器学习。这个框架是采用半监督式机器学习和深度包检测方法来分类SDN环境中的大象流。

Qazi提出了使用不同的机器学习算法来实现流量的应用意识。

通过评估，我们发现IP包地址和端口号不足以分类应用。受到上述研究的鼓舞以及对于解决SDN中多种路径能力不同的情况下路径高效域内路由的问题的需要。

AMPS旨在不使用深度包检测和多协议标签来分类流

三．应用意识的多路径流路由

AMPS引入机器学习算法的概念，对流量进行分类，并根据其QoS要求为每个流分配适当的路径。 这个想法是根据其优先级和类别来提供网络的受限资源，如带宽和低延迟路径。 图1显示了所提出的系统的系统图。 机器学习训练器（MLT）和机器学习分类器（MLC）被集成到SDN控制器中。 MLT用于训练分类器。 经过训练的MLC将分组分类为预定义的一组类。 根据所识别的类和策略，控制器为特定应用程序选择一条路径

1. 机器学习从流中获取应用的意识

机器学习采用监督式学习算法建立了C4.5决策树分类器。一个有40个特征的训练集作为输入。这个特征集合包括源目的ip，源目的端口，协议，前后向包总数，前后向总流量，前后向平均包长度，流的活动时间，流的闲置时间等。训练集包含了上述提到的特征的向量和相应的类标签。机器学习训练器会基于训练集产生一个模型。机器学习分类器由机器学习训练器产生，MLC获得特征向量的输入，预测给定的特征集所在的类别。C4.5采用了4个字段用于分类，他包括了分组间隔和包长作为主要的分类特征。DT的分类精度比其他的分类器更好。

1. AMPS控制器

AMPS控制器模块是SDN控制器模块的子模块。AMPS控制器包含时延检测模块（LD），可用带宽评估模块（ABE），链路开销计算模块（LCC）以及路径发现和选择模块（PDC）。LD模块计算连接到控制器的交换机之间的所有链路，ABE模块查看所有链路的可用带宽并且为每一条被添加进网络的新流更新可用带宽。一旦时延和可用带宽被计算出来了LCC就可以计算整个链路的开销。AMPS控制器有完整的网络拓扑并且会与MLC和MLT进行交互。刚开始的时候，AMPS控制器转发给MLT用于训练。如果训练被完成了，那么AMPS控制器就会捕获流的特征向量并且转发给MLC去得到相应的类标签。基于MLC返回的分类，AMPS控制器计算最好的从源节点到目的节点的K条可得路径，基于链路的开销和应用的分类来选择一个流。一旦这个路径被选择，AMPS控制器就会插入相应的流规则在那条路径所有的交换机中。

1. 网络中的数据包流

图1展示了网络中的数据包，这整个网络中有SDN交换机并且所有的交换机都被连接到了集中的控制器。网络中的数据包的流被描述如下：

1. 一个包被主机发送到SDN交换机
2. SDN交换机接收到数据包之后会在流表中匹配流规则。如果流规则存在那么数据包就按照流规则进行转发
3. 如果SDN交换机中不存在流规则，那么交换机就发送PACKET\_IN消息到AMPS控制器
4. AMPS控制器接收到PACKET\_IN消息后计算新流的平均长度并且发送FLOW\_MOD消息到路径上的所有交换机（不包括源交换机）。对于源交换机，一个PACKET\_OUT消息发送到相应的输出端口。因为源交换机没有任何的流规则。他继续发送PACKET\_IN消息到控制器。AMPS控制器收集来自同一条流前n个数据包的PACKET\_IN消息的特征向量，然后把所有收集到的特征发送给MLC。
5. MLC收到来自AMPS控制器的特征向量之后，就会把这个特征分到某一类
6. AMPS控制器收到来自MLC的类标签之后，就会查询PDC模块找到一条最好的路径，按照类的优先级。他发送流规则到选中路径上的所有交换机。新插入的流的优先级更高。我们假定，如果一个数据包已经被传输到了源目的地址的中部，此时一个新的流规则被插入了。这时，如果正在转发这个数据包的交换机已经安装了新的流规则，就按照新的路径进行转发，否则按照原来的路径进行转发。
7. SDN交换机在接收到PACKET\_OUT消息后，会转发数据包到各自的端口。如果接收到的是FLOW\_MOD消息，就会更改流表
8. SDN交换机转发包到目的地址
9. 控制器在第十个间隔，会通过发送统计查询消息给交换机来检查流的吞吐量，如果带宽需求被满足，AMPS控制器就会维护目前的这条路径。否则，这个操作就会在第六步被提出。
10. AMPS控制器安装相同的规则在已存在的路径，对于具有相同优先级的流。

四：实现

A.流规则计算

算法1描述了PDC模块中的多路径流路由，他收集来自不同模块的数据并且采用合适的决策。初始化时，AMPS控制器计算网络拓扑并且发送LLDP包收集链路统计信息。AMPS控制器创建了一个开销键值对，然后用LCC模块更新每条链路的开销。对于每条新到达的流，AMPS控制器都会收集特征向量并进行分类。CI是一个可变的存储在类中的值。YKSP算法被用于发现某条流从源节点到目的节点的K条路径。路径的带宽需求是从特征向量中计算。AMPS控制器迭代地从所有可得到的可用路径中找到能够携带流的可行路径的集合。一旦这个可行路径的集合被发现，Ci就会把所有可行的路径分配给所有的类。算法1显示了从源到目的之间前K条最短路径的集合。Pc是所有K条路径的开销。Pf是Pc中可行的路径。他会满足该类对于带宽和时延的需求。Nfp展示了可行路径的数据，Nc表示可用的不同类的数目。

B.应用分类

MLC模块实现了DT MLC。我们已经从我们的实验中获取了数据集，我们的实验有10个客户端，每个客户端都是用不同的应用，例如Skype, Facebook, Youtube and Dropbox。所有的这些客户端都会运行一个shell脚本，它会把处理id映射到相应的应用。并且在整个实验过程中也使用了wireshark进行跟踪。这个获取的特征集被作为机器学习算法的输入，朴素贝叶斯，朴素贝叶斯核估计，C4.5决策树与支持向量机。这些都是广泛使用的分类算法，通过实验发现，在此特定场景中，C4.5决策树得到了更高的分类精度，达到了98%。

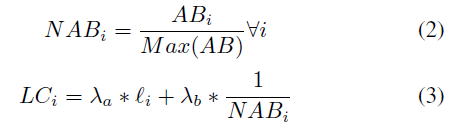
C．多流路由

1. 测量延迟：为了计算SDN网络中任意链路的延迟，使用了[6]提出的最优化算法。根据这样的一个理论，控制器将MAC地址作为广播目的地址来发送引用包。包的payload字段包含了包创建时的端口号和时间戳。报文携带一个PACKET\_OUT被发送到交换机然后命令这个交换机转发此数据包到特定的输出端口。当这个数据包达到下一个交换机的时候，由于下一个交换机没有响应的流入口，他会发送PAKET\_IN消息到控制器。在收到PACKET\_IN消息后，保持把源mac地址作为引用，控制器会查找ETT，ETT包括控制器->交换机1->交换机2->控制器.一个获取ETT(Conroller-Switch-Switch)的时间是发送STATISTICAL\_REQUEST给交换机，并且这个交换机会回复STATISTICAL\_RESPONSE消息，在时间Ts1内。用同样的方法可以得到Ts2。以此获取Latency(s1,s2)

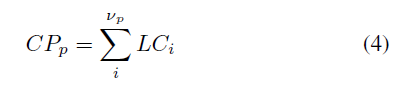
我们的时延检测模块(LD)有点不一样，并不是发送显示的控制报文去检查连接，我们使用LLDP包，他可以周期性的检查链路状态和链路延迟。这减少了发送到网络中的控制报文的数量。

1. 评估链路开销和给流安排路径

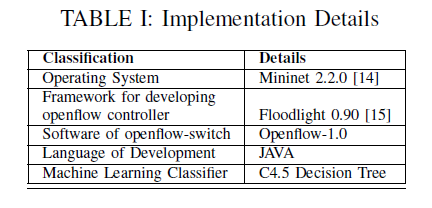
LCC模块负责发现每条链路的开销。链路的开销来自LD模块。LCC模块也使用辅助包评估了可得到的带宽。通过辅助包的大小和时延，可用带宽是可以计算的。可用带宽被用于查找这条流是否可以放在某条路径。YKSP算法被用于计算从源地址到目的地址的多条路径，YKSP的时间复杂度为Kn3。YKSP算法比大部分人的都要好。YKSP算法里面有很多的变量。但我们可以选择最基本的YKSP算法的实现。YKSP算法需要网络拓扑。所有的交换机和相应的链接在图中的格式。每条链路的花费LCi作为图中链路的权重。YKSP使用Dijkstra算法查找最短路径。如果最短路径被发现，他就会移除一条链路并试着再次查找最短路径，这个算法会迭代出K条路径。一旦第K条路径被发现，每条路径的开销就会被计算出来。链路的开销由链路的时延以及链路的可用带宽决定。时延是意思是说：特定的链路会造成多大的延迟，可用带宽模块是说:为什么这条链路会优先于其他链路。如果两个链路都有着相同的时延，那么链路的优先性就会由可用带宽来决定。可用带宽被正则化可以对带宽有一个更加公平的评估。



Li是交换机a和交换机b，λa和λb是相应变量的权重。一条路径p的花费，是链路开销的总和。



基于流的优先级和路径开销，一条流被发送到对应路径。我们使用hashmap的数据结构来维护链路信息。

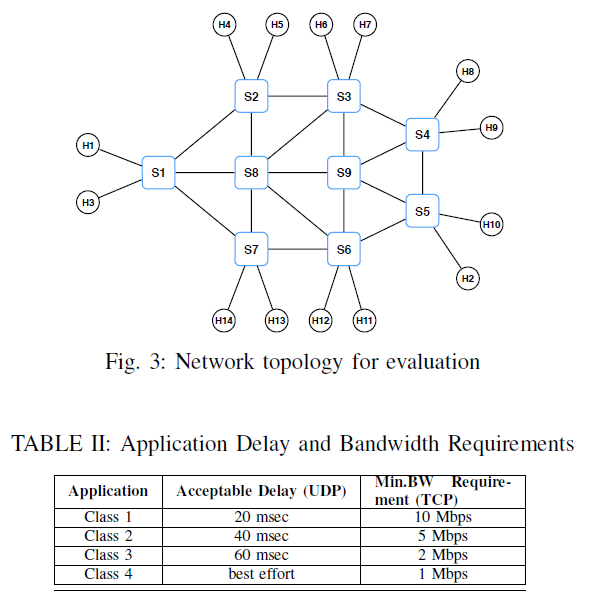


五：评估结果

1. 试验台的安装

我们监控主机H1和H2的性能，因为他们分布在网络的边缘。交换机之间的带宽32Mbps，主机和交换机之间的带宽1Gbps。四种类型的流：

Class1：实时流，Class2：文件流，Class3：web流，Class4：受限的文件传输流。期待MLC会根据流特征区分出这几种类型的输入流。



所有连接到AMPS控制器的交换机都会有一个专用链接，整个拓扑都搭建在mininet上，上述表示展示了可接受的时延和最小带宽需求，我们会从吞吐量，丢包率，抖动，时延等方面来评估。初始情况下，网络中会加载8条UDP流，作为背景流。需要观察的流在30s的时间内随机开始。参数的评估由混合的源目的节点对进行评估测试。流规则100秒内有效。观察网络参数的改变，如吞吐量，流过期时间。我们的实验中，流过期时间设定为1000秒，让我们有足够的时间去分析流的行为。

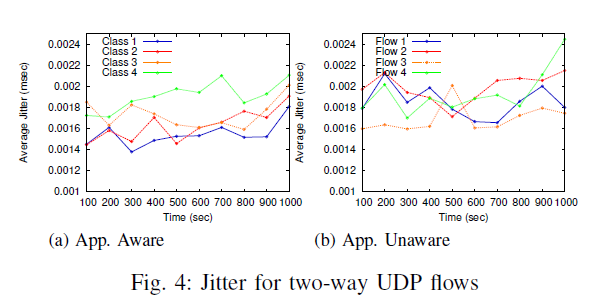
B.减少路径切换的影响

评估工作面临的一个问题就是大量丢包。当流过期之后，无论何时插入流规则，在旧路径中的交换机中的数据包都会创建大量的PACKET\_IN消息到控制器。忽略这些PACKET\_IN消息就会导致丢包。对于TCP这样的传输协议，控制器在安装新的流规则之后，并不能为这些缓存的数据包分配路径。所以，路径发现算法在流过期之前被调用，且新的流规则在过期之前被安装。这样可以大幅度降低丢包率，降低到0.001%。

C.实验

1）使用UDP进行抖动分析：

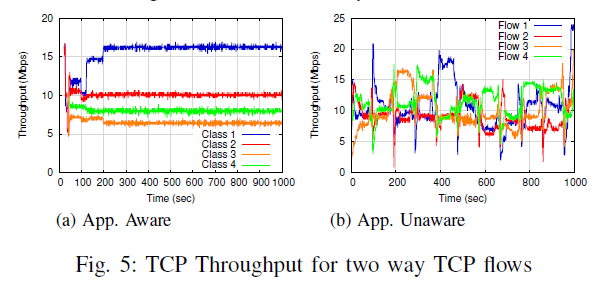
这个实验的目标在于验证号优先级的业务有相对较小的抖动。H1发送4UDP流到H2，H2发送4UDPflow到H1。背景流存在于所有交换机1和交换机5之间的链路。UDP被用作背景流。选择UDP作为背景流是确保骨干网络拥有恒定的负载。如果用TCP来当做背景流，目前的流和背景流就会相互波动，这是不可取的。不同流的延迟需求展示在表二里面。这里，我们计算了100秒内所有UDP流的平均抖动。图4a和4b显示了在UDP实验中观察到的抖动。X轴显示的是测量抖动的时间点，Y轴显示的是在那个时间内测量到的平均抖动。在图4a中，应用意识已经被考虑，路径是基于流的优先级进行分配的。在图4a中，我们可以观察到高优先级的类有着更低的抖动，因为AMPS控制器允许他们通过低时延的路径。可以看出，类1（最高优先级）相比类4（最低优先级）有着更小的抖动。在图4b中，没有考虑到应用意识，流的路由时基于最小跳数。因此没有对流进行分类，导致了高抖动。



1. 用TCP进行吞吐量的分析：

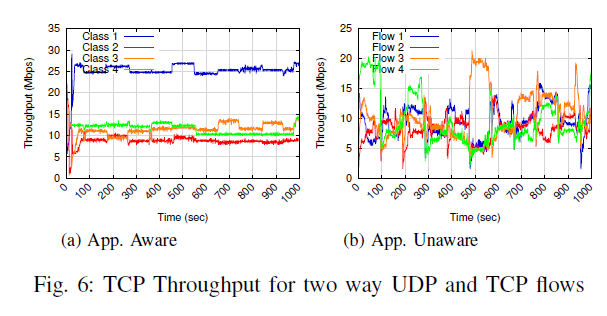
H1发送4个TCP流到H2，H2发送4个TCP流到H1。表二显示了各个流的带宽需求。每条流在0-50ms之间随机开始，伴随着一个大的TCP窗口，这个窗口会随着网络带宽而改变。根据AMPS的调控，类1流会被推到流量最少的那条路径以便于类1流的TCP窗口相比于其他流来说更高。这是因为其他流选择的路径会被背景流占用。这使得低优先级应用流的RTT会增加因此会使得窗口大小下降。在每90秒之后，控制器会检查每个流是否获得了所需的吞吐量，如果没有，这条流就会被重路由。由于在每90秒的路由决策都是基于流量统计的，在流过期之前，新的被安装的流规则使得流不需要遭受更多，因为TCP拥塞避免机制。这是TCP吞吐量几乎没有波动的原因。这一点在图5a中被提及，流4比流3得到了更大的吞吐量，因为类3和类4都对他们的带宽保证很满意，因为我们没有指定任何的带宽上限。因此，这些流不会被更改。如果这些流的需求比最小带宽更大，那么他们将会被重路由。

图5b描述了相同配置下的实验结果，但是在这里面流没有被分类（没有应用意识，像4b一样）。因此，他们没有被路由到正确的路径，并且每条流都会互相影响相同路径上的其他流。在每100秒过后，每条流都会过期，并且被重新放置到一个随机的路径。

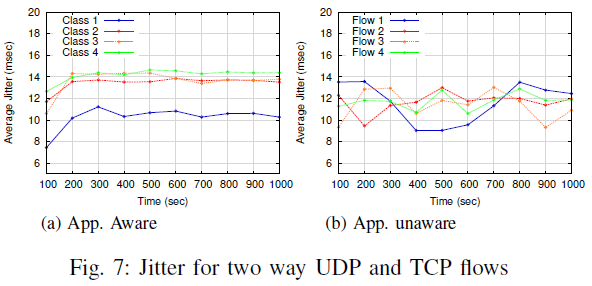


1. 用TCP和UDP来进行吞吐量的分析

在这个实验中，我们观察系统在异构网络中的行为。评估采用了和上面章节类似的设置。包括4条TCP流，4条UDP流，H1->H2,H2->H1。但是吞吐量仅仅通过TCP流来计算。

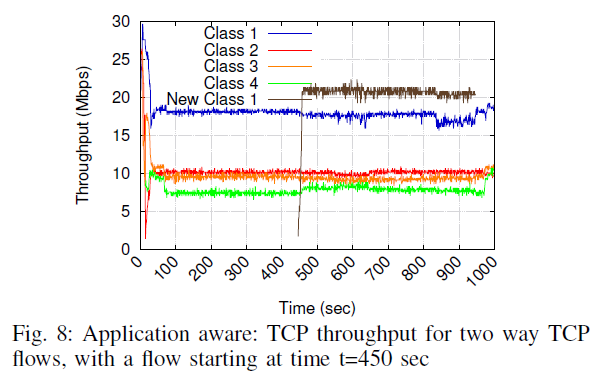


在图6a中，类1获得了更高的吞吐量，剩下的流的带宽需求被满足，所以不会被重路由。图6a相比于图5a来说出现了波动是因为图6a里面既有TCP流又有UDP流。UDP流路径的变动会影响到TCP吞吐量。在图6b中，流被重路由，因为没有基于分类需求，因此他们没有一个清晰的类吞吐量。



4）用TCP和UDP进行抖动分析：

评估设置部署和上面一样。在这个试验中，抖动率仅仅通过UDP流来计算。当网络中负载较重时，基于应用意识能够带来清晰的好处。因为类1流只能忍受很小的抖动和时延，当网络具有流优先级意识的时候，类1相比于其他类来说有着非常低的抖动。



1. 当有一个新流在t时刻开始时，用TCP进行吞吐量分析

在图8中，类1流在t=450sec时开始。网络中承载着TCP和UDP流。当一个新流进入到网络，AMPS控制器应该找到最好的祖静分配给流1,。在图8中，被证明AMPS控制器为类1流找到了最好的路径因此吞吐量很高。

六．结论和未来工作

为了让企业网中的不同应用的不同需求更加便利，我们介绍了AMPS控制器，使用了ML在SDN中。AMPS主动分类进来的流量并且应用了相应的QOS策略。每条流选择最合适的路径进行路由，以期望获得更低的时延和更高的吞吐量。与MPLS不一样，在我们的模型中，没有新的tags添加到数据包上。这个想法通过在mininet上进行不同的实验来进行评估。结果证明，把服务分成不同类别的应用，可以显著的降低时延和提高吞吐量。很显然当存在多条路径可以安排的时候，为分类的应用流无法满足要求。这也强调了，基于网络应用分类可以提高用户体验。