**基于流持续时间的带宽分配方法**

**技术背景**

现在很多对延迟敏感的应用（delay-sensitive application）部署在数据中心（Data Center，简称DC）中，部署在数据中心的应用需要数据中心网络需提供低延迟和高带宽。根据亚马逊公司的统计，传输每增加100ms，就会造成1%的利润损失。

特别的，许多应用的流同时在数据中心中传输。这些数据流，有的需要更多的带宽，有的需要更低的延迟。传统的网络传输协议TCP，在满足流对延迟和带宽需求上，性能存在一些不足。首先，TCP使得交换机的缓冲区队列很长，这样，会导致很高的排队延迟，从而使得网络传输的延迟大大增加；其次，TCP因为出现拥塞信号时，窗口会减半，着就会导致网络链路震荡严重，使得数据中心网络链路利用率低。针对TCP在数据中心网络中传输的缺陷，DCTCP被提出。DCTCP是根据网络的拥塞程度进行拥塞和速率控制。DCTCP可以降低交换机的排队延迟，并且使得链路利用率提高，但是DCTCP是一种对数据流公平分配带宽的策略，不能够根据数据和和应用的特征对带宽进行分配。针对DCTCP的这些不足，业界提出两类方法：

1. 给流设置期限（deadline）。这种方法是，加入期限因子，对流进行带宽分配。在本类方法中把数据流根据流的大小（flow size）和期限（deadline）的一个函数对数据流进行区分。紧急程度高的流，获取更多的带宽，紧急程度低的流获得的带宽少，这样，更多的数据流能在截止时间之前完成。这类方法的代表主要有D2TCP，D3和LPD等。
2. 此外，还有一类的方法是只根据数据流大小（flow size）进行带宽区分。因为数据中心中，往往小流的紧急程度要比大流的紧急程度高，因此，给小流高优先级，也能让更多紧急的业务更快的完成。这种方法的代表主要有L2DCT，pFabric等。

根据当前业界的方法，我们可以归纳以下结论：首先，无论是设置期限的方法还是只是根据流大小进行带宽分配的方法，都需要得知流的一些特征信息（flow size 或者deadline），然而，这些特征有的时候在数据中心中是很难被得知的，因此在实际使用中，这些方法往往有一些局限性。其次，第一种方法，本质上是在降低数据流的延迟，其实是满足了数据流对延迟的需求；第二种方法，是让小流获取更多的带宽，降低流平均完成时间，本质上是让流获得更多的带宽。当前业界的解决方案，只是尝试满足其中的一个要求，并没有同时满足这两个要求。因此亟需一种方案，来同时满足这两方面的需求。

**发明内容**

本发明提出一种基于流持续时间的网络拥塞控制算法。

对于数据中心中混合流量的场景，本发明首创把使用流完成时间（durtime）引入到带宽分配中。一条数据流随着流持续时间的增长，初始的带宽很大，随着时间的流逝，带宽逐渐的变小。这样做的优势是，既能够使得对延迟敏感的流获得更多的带宽，同时能够使对带宽敏感的流获得更高的带宽。因为，对于期限很近的流，使用流完成时间（durtime）进行调整，能够使得这些流开始获得更多的带宽，从而在期限（deadline）之前完成。其次，对于平均带宽要求多的小流，因为开始时，流的优先级最高，因此，获得的带宽更多，因而，平均带宽也大。所以，使用流完成时间，既可以满足数据流对延迟的要求，又能够满足数据流对平均带宽的要求。

本发明中流的持续时间定义为，当前的时间－流开始发送的时间，即为

定义紧急因子d，d越大，意味着根据期限调整的力度也越大。当流持续时间大于懒惰时间（threshold\_lax）时，因子d，为最高优先级（DMAX），当流持续时间小于紧急时间（threshold\_tight）时，因子d为最低优先级（DMIN）。计算紧急因子d的算法如下：

d=DMIN

else

d=DMAX

根据网络的拥塞程度和紧急因子d，本专利提出FDRC (flow duration time rate control)协议，FDRC协议有三部分：

1. 交换机。交换机要启动ECN功能，并且，设定一个阈值K，当交换机的队列大于K时，发送给接收端的数据包就会被标记CE。
2. 接收端。接收端查看从交换机发送来的数据包是否标记了CE，如果标记了CE，那么给发送端的ACK数据包，就会标记ECN位，否则，这个标记为置0。
3. 发送端。发送端首先统计本个RTT内发送的数据包被标记ECN位的数据包的比例F：

然后根据F计算拥塞程度，的计算方法如下：

g是权重因子

根据拥塞程度紧急因子d进行窗口调整因子f的计算：

和TCP，DCTCP等不同，FDRC与LPD类似，同时修改窗口增加和减小部分，窗口变化函数如下：

**实施实例：**

图1是根据本发明实施例一的基于滑动窗口的带宽分配系统的结构示意图。该系统中包括汇聚设备和工作机，为托管的不同应用程序服务，数据中心中的应用例如，搜索，购物，银行计算，证券计算的业务都是部署在这种模型体系下的。在具体的环境中，请求从顶层的汇聚设备发出，然后到达下面的汇聚设备和工作机器，下面的汇聚设备和工作机器，按照上面的的指令进行计算，计算完成之后，把结果反馈给上层的节点。

在数据中心网络这种应用环境中，汇聚设备可以为交换机以及/或者路由器，工作机典型的为各种服务器。工作机把结果反馈给汇聚设备时，会有延迟问题，如果延迟过高，那么就会影响服务的性能，进而对用户体验造成不利的影响。因此，需要尽可能的降低数据在各种设备之间传输的延迟。

在数据中心中，很可能很多应用并发的在其中运行，因此，它们会共享网络资源，因为不同的应用的需求不同，因而对网络提供的性能要求也不同。比如，有的应用需要低延迟，有的应用需要高带宽。当前的协议，比如DCTCP，D2TCP，L2DCT只能满足一部分应用数据流的需求，因而存在一些缺陷。



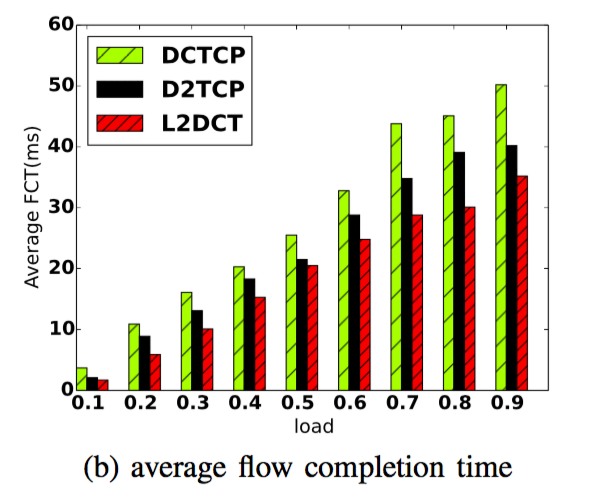
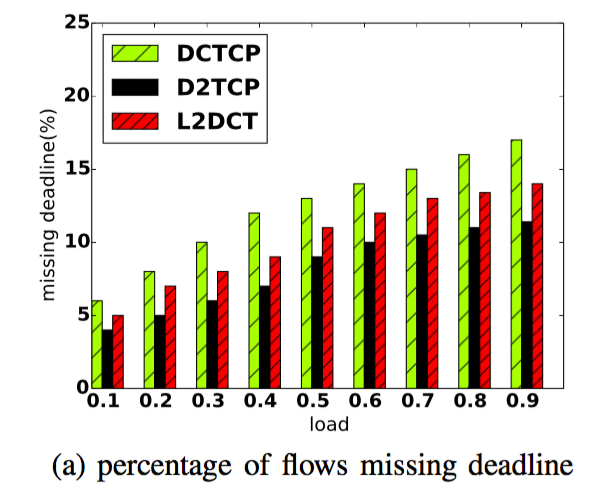
**图1: 数据中心汇聚－分散模型**

首先对当前业界方案的不足进行说明。

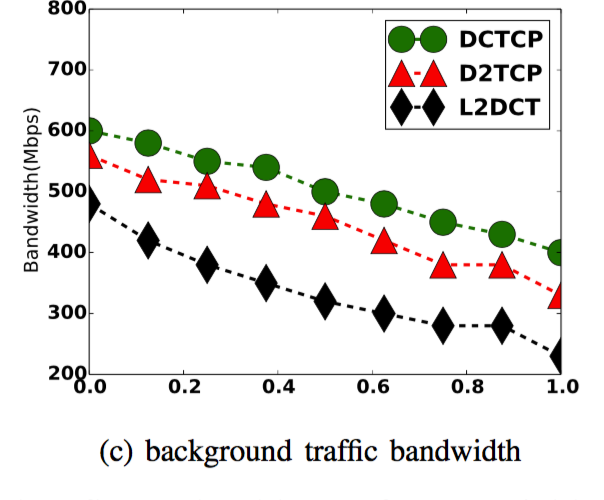
启动10个工作机器，10个工作机器同时往一个接收端发送信息。所有的链路的带宽大小为1Gbps，设置交换机中队列的阈值K＝25。在这10个工作机中其中2个发送背景流，另外的8个工作机发送流大小在100KB到1MB之间的数据流。把这些流的集合平均分成两份，第一份，设置发送的数据流的期限(deadline)为30ms，第二部分，尽可能多的减少流完成时间。比较DCTCP，D2TCP和L2DCT的性能。图2为这几种方法的结果。

从图2（a）可以看出，D2TCP和L2DCT在流错失时间比例方面比DCTCP要好，并且D2TCP要比L2DCT中流错失时间的比例小。从图2（b）可以看出，对于流完成时间，L2DCT的表现要强于DCTCP和D2TCP。图2（c）可以看出L2DCT从背景流中抢夺带宽的能力要强于D2TCP和DCTCP。

综合图2的结果，我们可以发现，D2TCP在流错失时间，也就是优化流延迟方面的表现要强于DCTCP和L2DCT。L2DCT在优化流完成时间，也就是带宽方面要好于D2TCP。这两类方法，只能满足一方面应用的需求，但是无法满足所有应用的需求。



**图2（a）流错失时间比例 图2（b）流平均完成时间**



**图2（c）背景流带宽**

**实施实例：**

启动两条数据流，flow1和flow2 ，两条流的带宽分别为R1 和R2，假设拥塞窗口从wmin 变到wmax，这个过程中，窗口的平均大小为

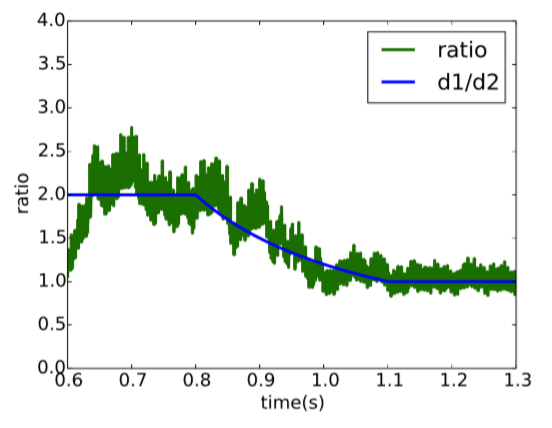
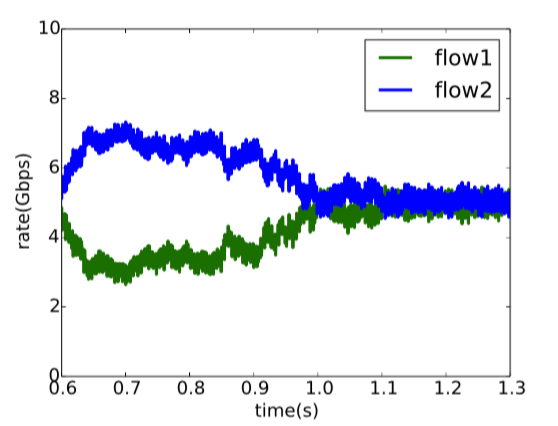
因为在速率控制中，窗口是按照锯齿状进行增大的，因此，有下列关系：

使用w1代表最小的窗口大小，使用w2代表最大的窗口大小，R1 和R2代表两条流的速率，根据窗口和带宽的关系，有如下结论：

当d1和d2很小的时候，得到下面的结果；

因此得到结论，对于二条流的系统，当期限d1和d2很小的时候，数据流的带宽和期限成反比。

为了验证我们的结论，在ns-2系统中，启动两条数据流，第一条数据流的启动时间是0s，第二条是0.5s。设置阈值threshold\_tight=0.3s, threshold\_lax=0.6s。设置DMIN＝0.05, DMAX=0.1。图3（a）显示的是两条数据流的带宽，图3（b）是两条数据流的带宽比和期限比。 从图3（b）可以看出，当时间在0.6s到0.9s之间时，两条流的带宽比基本和期限的设置成反比，并且随着时间流逝，第一条流的带宽逐渐变小，最后在t=1.0s之后，两条流的带宽基本相等。



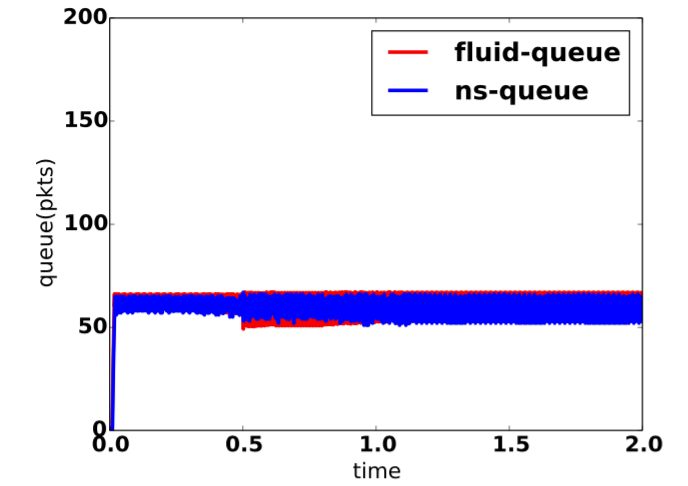
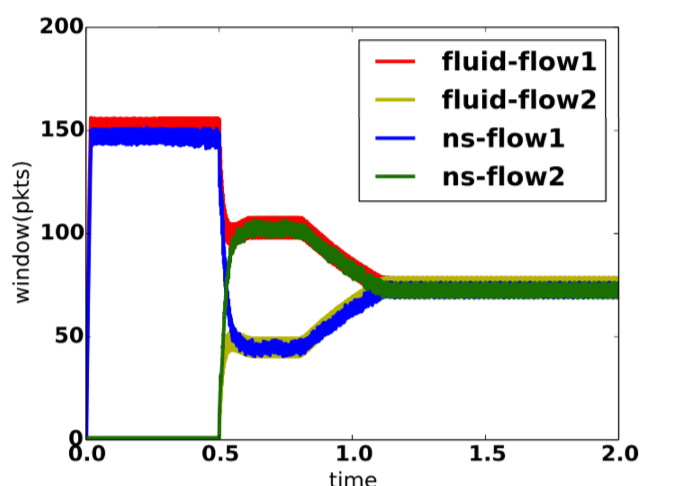
**图3（a）两条数据流带宽 图3（b）两条数据流带宽比和期限比**

图3的实力结果和理论的推倒基本结论相同，即当期限很小时，对于两条流系统，数据流的带宽和数据流的期限成反比。

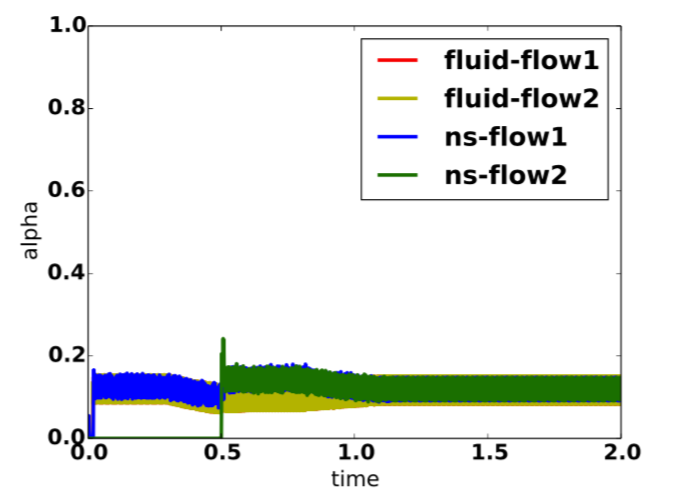
**实施实例：**

根据fluid model对FDRC进行如下所示模型：

前两个公式描述的是FDRC交换机根据拥塞程度对数据包打标记的过程，第三个公式描述的是交换机队列的长度，第四个公式描述的是滑动窗口的变化过程。这个动态模型描述的是FDRC的控制过程，根据模型可以推算FDRC的特性。为了验证模型的准确性，本实例对用模型推导的窗口，队列和拥塞因子变化和实际仿真结果进行验证。



**图4（a）数据流滑动窗口对比 图4（b）队列长度**



**图4（c）**

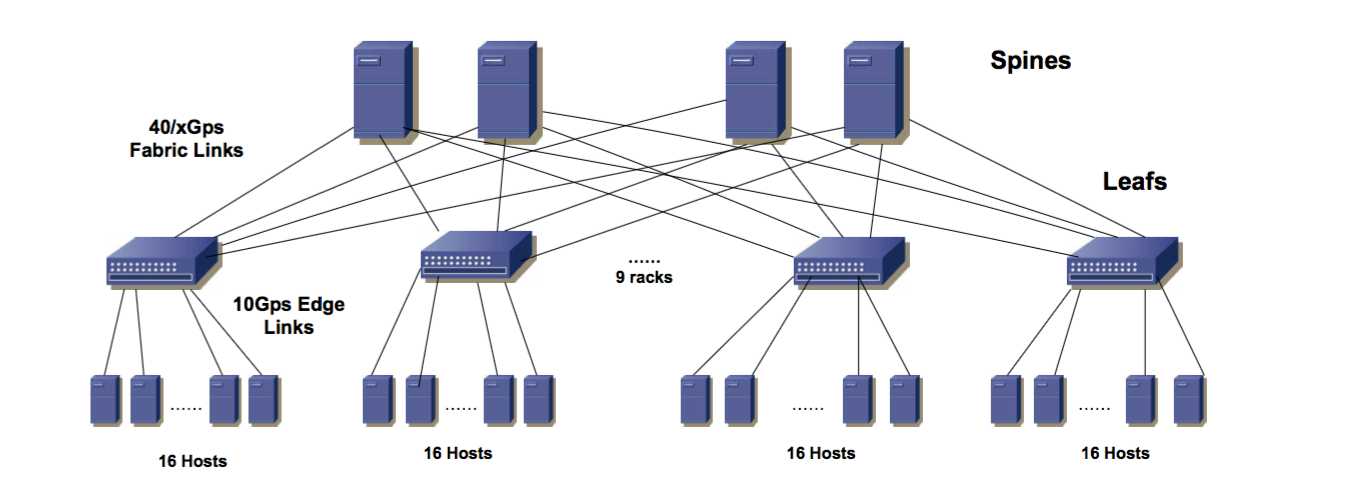
在验证实例中，首先启动两条数据流，两条流的启动时间分别为0s和0.5s。设置DMAX＝1，DMIN＝0.5，并且设置两个阈值，threshold\_tight=0.3, threshold\_lax=0.6.得到的对比结果如图4所示。

图4（a）是数据流滑动窗口的对比，可以发现，模型推导的结果和使用仿真结果基本可以拟合。图4(b)显示的是队列的长度，可以发现，队列的长度的误差在5以内。图4（c）是拥塞程度的对比，可以看到拥塞程度基本能拟合。根据试验结果我们可以发现根据模型推导的窗口，队列和拥塞程度变化和仿真结果基本能拟合，因此可以用模型来推倒FDRC各个参数的设置。

**实施实例：**

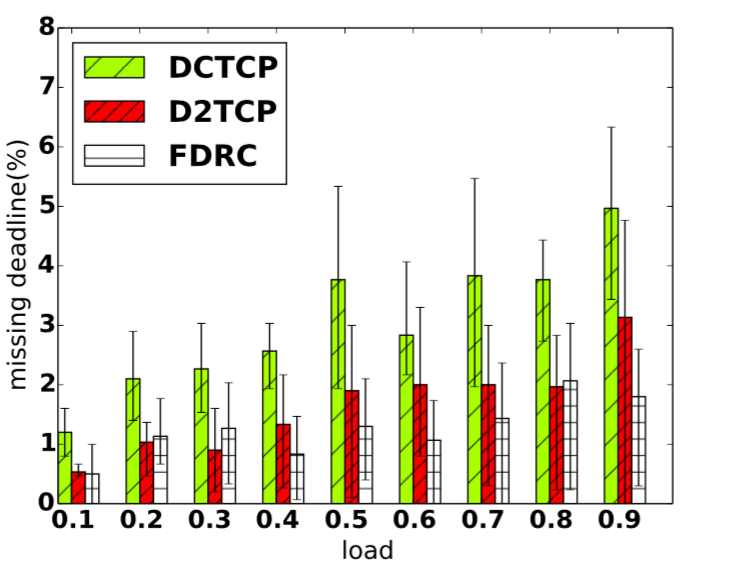
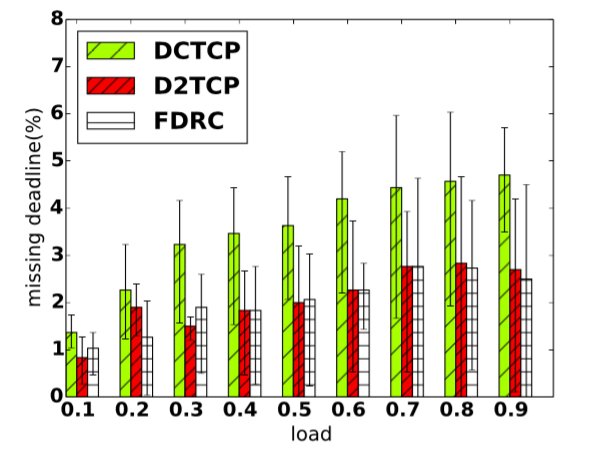
和L2DCTP相比，FDRC可以在同时减少流完成时间的同时进而减少错失期限的流的数目。和D2TCP相比，FDRC在减少流错失期限的同时，能够减小流的平均完成时间。为了验证FDRC的性能，在NS－2中，我们搭建了如图5所示的数据中心拓扑。

图5 为spine-lieaf 的数据中心拓扑，这种拓扑是一个fat-tree类型的拓扑，在这个拓扑中，leaf 交换机和每一个spine连接，同时，每个leaf 交换机下面挂载了16个服务器。Root 交换机到leaf交换机之间链路带宽是40Gbps,leaf 节点到下面host的带宽是10Gbps.

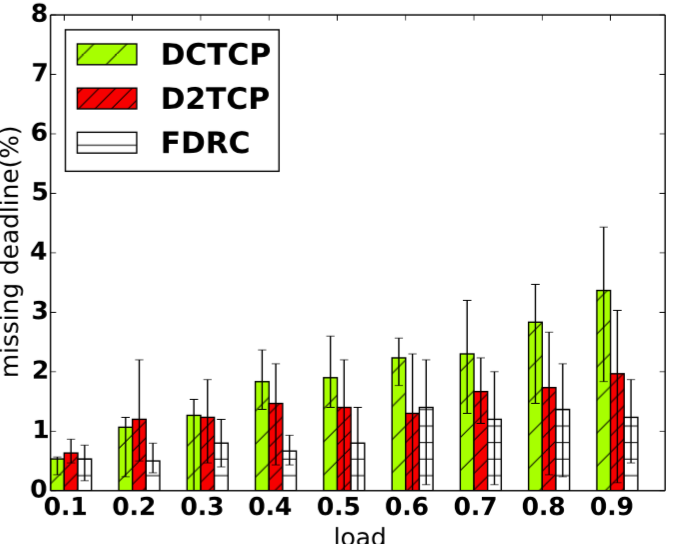
****

**图5: spine-leaf 数据中心拓扑**

在本实例中，使用数据中心的网络搜索流量模型，数据流的期限设置服从指数分布，并且只有一半的数据流有期限，期限设置有三种，分别为紧急期限，平均大小是30ms，温和期限，平均大小为40ms,长期限，平均大小为50ms。变换数据流的期限，得到如下所示的结果。



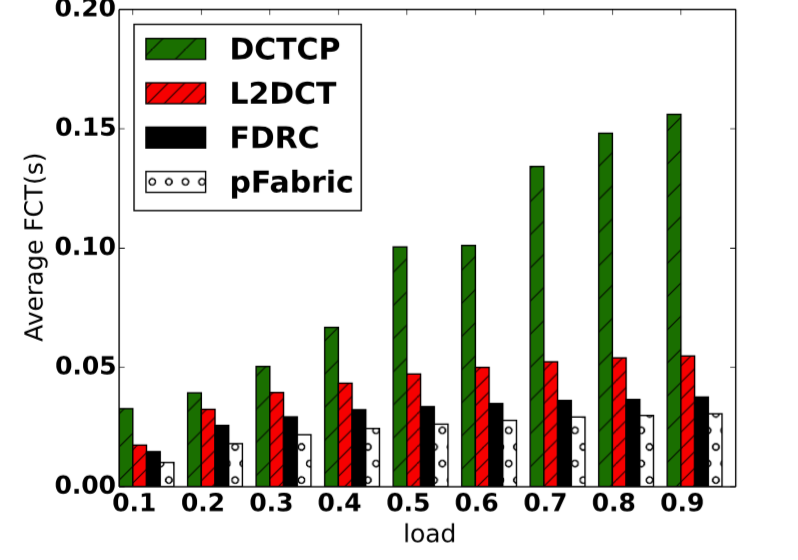
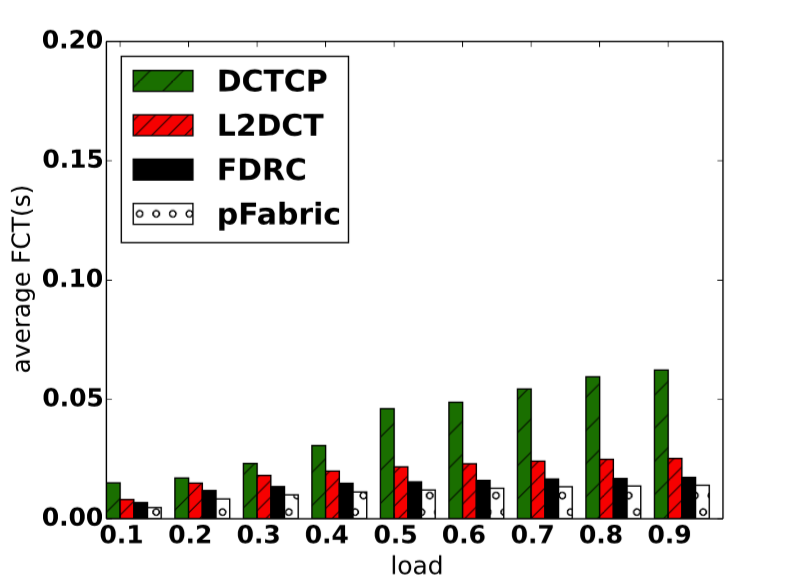
**图6（a）紧急期限(~30ms) 图6（b） 温和期限 (~40ms)**



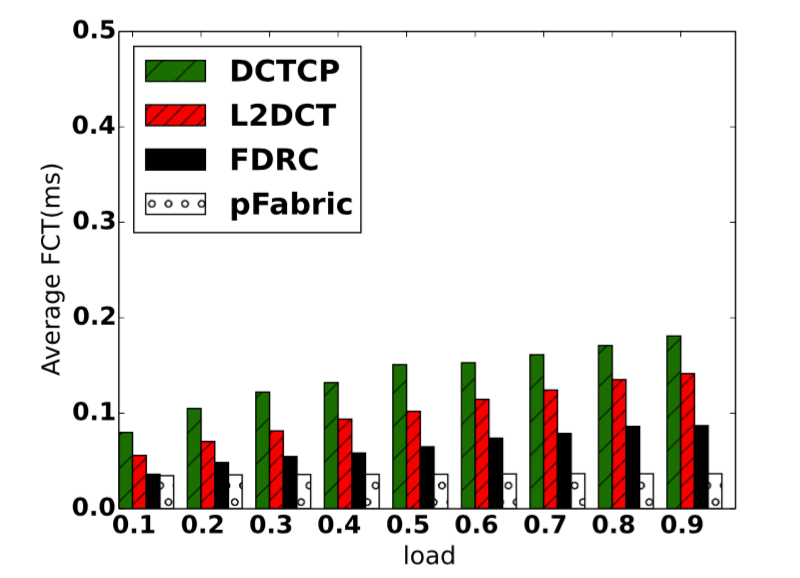
**图6（c） 长期限 (~50ms)**

图6（a）是数据流的期限设置为紧急期限的场景，发现DCTCP错失流期限的比率最高，D2TCP和FDRC的表现相当，图6（b）是温和期限的场景，发现FDRC的表现要由于DCTCP和D2TCP，图6（c）是长期限的情形，结果和温和期限的结果类似。因此FDRC可以有效的减少流错失期限的比率。

在数据中心中心中，使用网络搜索流量，所有的数据流没有期限，图7是流完成时间的对比。



**图7（a）平均完成时间 图7（b）大于500KB流完成时间**



**图7（c）［0，100KB］流完成时间**

图7(a)是流平均完成时间对比，可以看到pFabric的表现最好，其次是FDRC, L2DCT,DCTCP。图7（b）是流大笑大于500KB时的场景，图7（c）是流大小在0到100KB时的场景。可以两种场景的结果和流平均完成时间的结果类似。可以发现FDRC的表现不如近似最优策略pFabric，但是好于L2DCT和DCTCP。因此FDRC能够有效的减少流完成时间。