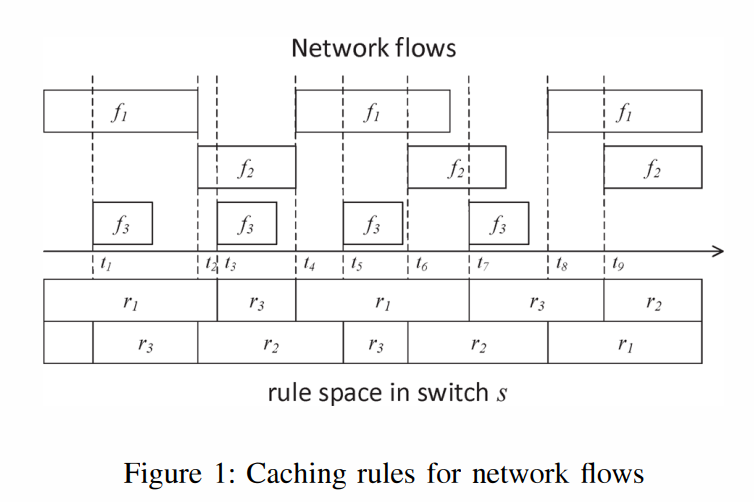
SDN 流表查询中的cache加速技术调查报告

小组成员及分工

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 |  |
| 张豪 | 2017213815 |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. SDN流表查询技术背景介绍

在找到新的规则替换TCAM中缓存的规则时经常会采用FIFO或LRU算法。但这些方法都有一系列的问题。对于FIFO算法，强调先进先出，有些很少用的规则反而在TCAM中停留了太长的时间，对于LRU算法也只是考虑了packet而没有考虑flow。



例如在上图中，TCAM使用LRU来更新cache，由于不知道每个包的后续情况造成cache频繁抖动，整个过程的命中率为0。

1. cache替换策略

1. 性能（命中率）

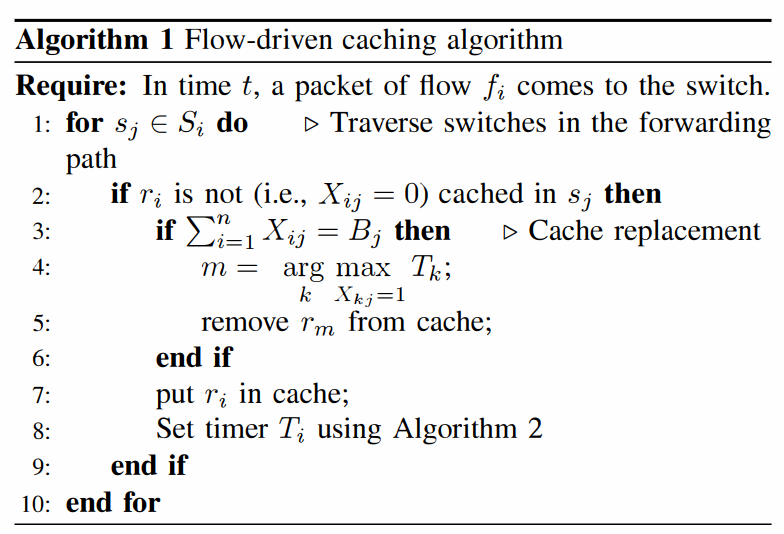
2. 可靠性（缓存的一致性）

3. 稳定性（时延抖动）

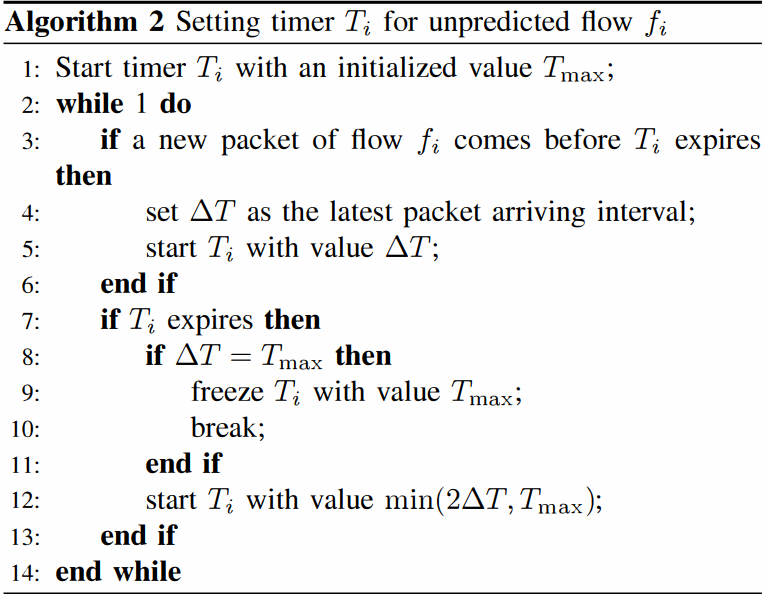
基本思想：

1. 如果缓存一条流的规则，那么整条路线上都要缓存这条规则
2. 每个流会计算一个计时器T预测下次缓存命中的时间，缓存替换时把预测时间最长的规则替换出去

**第一步**：缓存一条流的规则的时候这条流涉及线路的所有路由器都要把规则添加到cache中



**第二步**：给每一条规则涉及一个计时器T，如果cache满了就把T最大的规则替换出去。T代表这条流的下一个包的预期到达时间。对于可以预测的流，可以把T直接设置为下一个包的预期时间，对于未知的不可预期的流，则需要使用算法预测T，预测的算法表述为：



这个算法的意思就是先设置一个最大值，然后计算下一个包到达的间隔设置为T，到了T时间如果没能hit就把T设置为2T（但不超过最大值）。

1. 对生命周期短、负载量少的流的处理方法

四、cache的更新问题