# 1系统模型和设计

## 问题建模

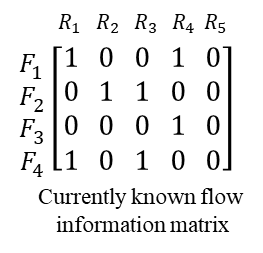
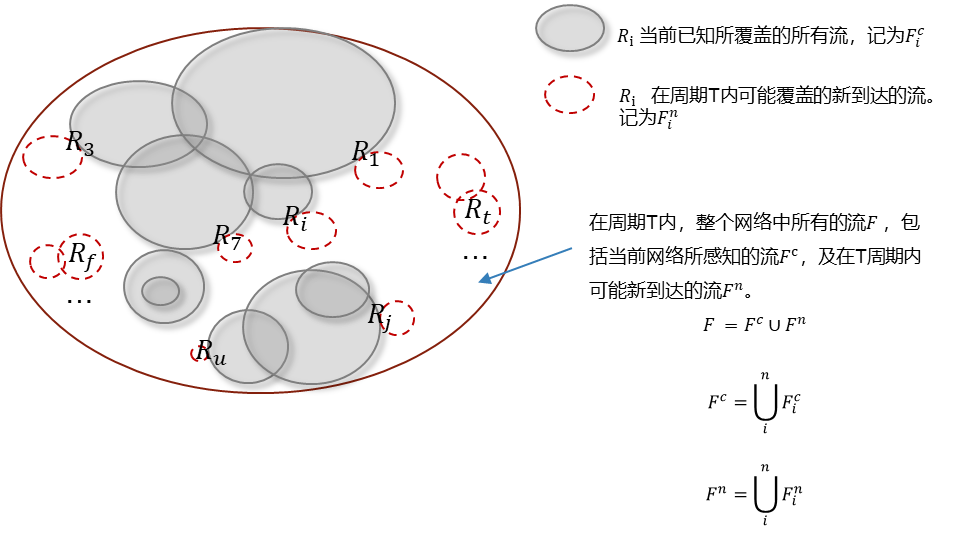
无论在SSP采样模式中还是在RSP采样模式中，采样的最终目标就是为了在给定策略时间内，收集到足够多或者满足需求的流数据。在SSP采样模式中，体现的就是在一个采样周期T内，为一个或多个节点分配一定的采样时间，而进行的策略。在传统的方式中，通过简单的影响力评估，或者静态的策略，并设置固定的采样时间，来建立起基础的采样系统。然而对于复杂的网络环境而言，尤其是大型网络中的流量变化大。而对于采样而言，要求实时性，精确性，和高效性，低入侵性以及满足采样收集器的容量范围之内等基本要求。[1]中，通过社交网络的方式，对节点基于流的中介中心性对节点进行了动态影响力量化，并选取Top K的节点进行等采样率采样，该策略很好的体现了策略对于实时流量的响应能力，但该基于影响力的评估过于单一，而且并未考虑到可能的新到流的情况。因为网络中的MiceFlow十分的多[4]，因为会存在漏流的情况。（强调对于流量采样问题，前期的工作中并没有对该问题进行建模，而大部分使用的是社交网络的知识或者基于SDN的相关技术）。

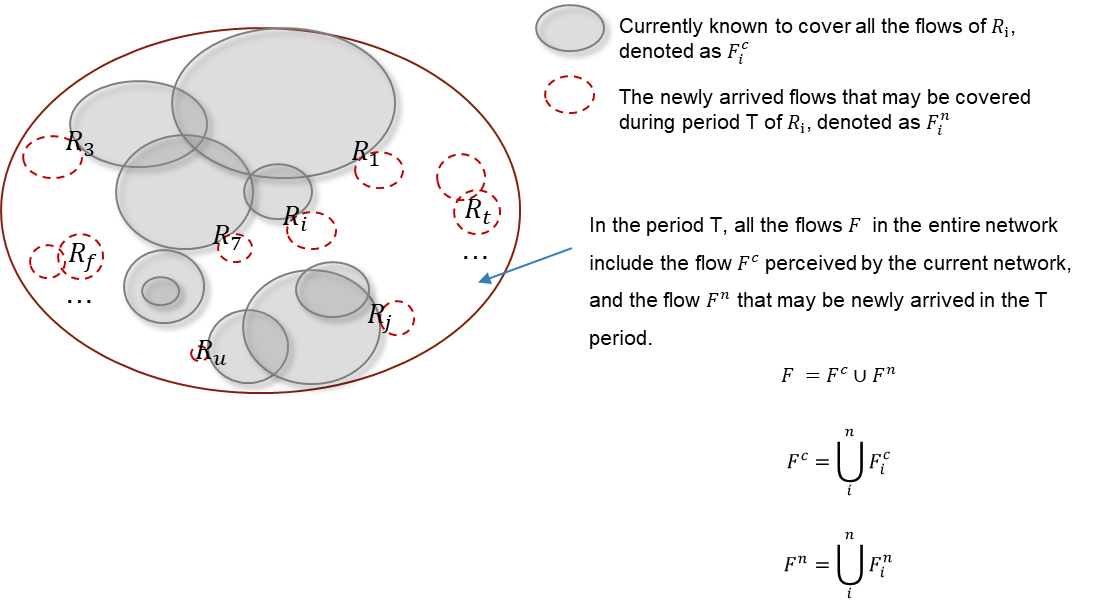
我们从自适应、敏捷性、准确性、高效性、协同性这四方面入手，对问题进行抽象和建模。构建一个AAA的Co-Sampling Model。我们通过一个全新的视角去抽象这个问题，使得该问题更加直观，并对该问题进行建模。

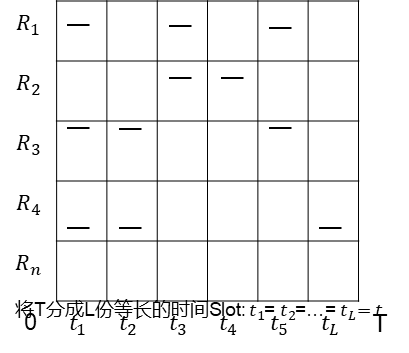
我们认为，节点之间的采样存在协作关系，这种关系是多维度的，体现在他们之间的相互约束关系，在当前网络中流的覆盖关系，topology的中静态关系等。

Fig.1展示了我们对问题的直观抽象。对于一个采样周期T内，整个网络中的流集合F正如一个被红色轮廓描绘的区域，而对于R1..Rn而言，每一个节点都覆盖了0条或多条流，记为集合，正如流量信息矩阵所展示的那样，在Fig.1中为灰色的阴影区域。红色虚线区域，则为节点Ri 所对应的在周期T内所可能会涵盖的新到达的流。灰色区域之间的重叠区域则为各节点之间所覆盖的流的重叠部分，。红色虚线区域之间的重叠部分则为各节点之间所覆盖的新到达的流的重叠部分。有的节点既包含灰色区域，又包含红色区域；而有的节点只包含红色区域或者灰色区域，这代表了各节点在该周期内的价值。覆盖的面积越大，则该节点覆盖的流越多。因此，Flow-Level的采样问题，可直观地将其转换为面积覆盖最大化问题。即：在给定的收集器处理能力和其他约束条件下，实现为各节点的采样时间分配，从而使得覆盖的面积最大化(流覆盖数量最多)。

在该问题定义下，我们提出一种基于Slot划分时间的量化模型。Fig.2 展示了这种模型的方式。首先我们将采样周期T分成L个等长的Slot；对于每一个Ri而言，需要确定它的采样Slot集合 。对于任意一个交换机而言，一个Slot时间内它能产生的价值表示为，这里直观的表示为采样到流的数量。而对于所有节点而言，他们被分配的Slot数量和次序都是需要计算出来的，其原因是，Slot的数量对一个节点所产生的价值是正相关的，分配给某个节点的时间越多，采样到流更多的期望值更大，而每个节点的Slot的次序则隐含的表达了节点间的相互约束：因为两两节点间含有重叠的流，而在最大化覆盖面积时，应该去除在同一个Slot下各节点间重叠面积的多次累加。对于Ri所产生的价值而言，其量化除了考虑在一个单位Slot时间内，所采样到的已知流数量的期望值外，还应该考虑在周期T内可能到达的新的流的数量，但对于未知流而言，即使我们通过一些流的到达分布模型来量化出到达某台交换机的数量，但是却不能知道各个节点间的到达的未知流之间的关系，不可预知未知流的路径和节点间重叠流的关系，因此很难量化在周期T内新到达流对节点间的约束关系。然而对于一个节点而言，它的静态Topology影响力和短期内活跃度，可以用以近似衡量它在周期T内所带来的价值，他们代表了，在周T内节点在全部节点中的一个影响力，影响力越大，覆盖更多的流的可能性越大。而且这两个属性是属于节点私有的，与其他节点间不存在约束关系。把节点在单位Slot内覆盖的已知流的数量/流的总数量，则为单位时间Slot内，节点所带来的对于当前流的覆盖率，我们称之为基于流的动态影响力。







我们给出下面的优化模型，即影响力最大化模型(Maximize Influence)(公式)，它是基于流量覆盖问题的一个转换：利用社交网络中的量化模型，将流的覆盖条数转换为覆盖率，来量化该节点在单位t内的动态影响力；用Si 和 Hi 来量化节点在T内的影响力。因此，在t单位时间内，某个节点的影响力应该总和考虑他们三者，我们使用加权的方式来得到节点的综合影响力值。

其中Di的量化中，我们假设了流fi 包的到达强度服从强度为λi的泊松分布。

其中满足如下条件的约束：

; 其中

对于最大化影响力问题模型中，充分考虑了已知流量的覆盖，未知的新流覆盖，节点的选取，以及时间Slot的分配，和Slot时间序列的安排。而时间Slot的安排中，因为节点间的重叠流在同一个Slot下会产生冲突，使得降低了整体的覆盖率，因此为得到最优化的解，越是重叠数量大的两节点，若他们都分配有大于0个Slot情况下，他们重叠的Slot数量应该尽可能少，这又同时解决了另外一个重要的采样问题，即：如何降低重复包的问题。因为各节点若包含相同的流时，为使得结果最优化，会在Slot上面相互排斥，因此最优化的解中，每个节点的的序列能够使得在整个周期中，重叠采样所覆盖的面积最小化，因此能够极大的降低包的重复率，同时保证最大化的采样精度。因此，节点间是协作的，通过相互间的约束关系，流的覆盖关系，静态Topology关系，活跃度等作用在一起，最终最大化采样的流数。(在采样过程中，多个节点采样时，会产生大量的重复包，因为流可能经过任意多个节点，而同样的包会在多个节点被采集到，这样不仅浪费了宝贵的采样资源，也同时降低了采样精度。尽可能降低重复包，通过节点间流的叠加关系来避免该问题，不仅可以提升采样精度，更可以提高收集器上层应用的效率)。

模型(1)中综合了三个优化问题：节点选取(Slot < 0 或者 Slot >= 0)；Slot个数分配；Slot时间序列的确定。模型(1)包含了这三个优化问题的约束条件以及他们之间存在相互的关系及约束。我们在量化的同时考虑到复杂性和可行性，实际上很难直接求解该问题的解。因此我们把该问题模型分解成三个子模型：节点选取，Slot个数分配，Slot序列的安排。每个子问题，我们对其进行独立建模，并使用独立的算法来求解最优解或近似最优解。最终能够有效得到该问题的近似最有解。

该问题不仅是多重背包的变种，在多重背包的问题上，添加了足够多的约束条件。而要求解该问题，是一个NP-hard问题。因此，我们将其分解成三个部分来近似求解该问题。

节点选取：

根据模型（1）的定义，选择K个影响力最大的节点，作为采样节点，并分配Slot。

在最优化模型(1)中，选择采样节点和分配Slot个数以及决定各采样节点的Slot次序他们之间的约束是各节点间存在已知流的重叠问题，因此要独立分隔这三个问题，首先需要将各节点间的流的重叠问题给去除。我们采用模型(1)中的节点影响力量化方式，通过动态影响力、静态拓扑影响力、节点历史影响力的三者来综合量化各节点并选择前K个影响力最大化节点，并每一轮(共K轮)的选取过程中，采用高影响力节点私有化重叠流的方式，解决流的重叠问题。而此时的动态影响力Di表示了该节点基于流的中介中心度影响力。Dik表示在第K轮选取节点过程中，选择第k个节点时，节点i的动态影响力，在第K轮选取中，若i节点综合影响力最大，都会私有化它所包含的所有流但不包含1-k-1轮中的节点所包含的流。

公式(15)给出了综合影响力的量化公式。

时间Slot分配：

节点被选取之后，需要为每个节点分配Slot个数。这是一个典型多重背包问题：共有K类待选物品(Rs1-Rsk)，每类待选物品都有T/t个(T/t个Slot可以选择)，而每件商品在一个单位Slot t内的收益是V(Ri,ti)，代价是W(Ri,ti)。因此，我们可以采用求解多重背包的算法来为每一个Ri分配Slot个数。V(Ri,ti)是模型中的部分，取|S|为1。而W(Ri,ti)= 单位t时间Ri的包速率pkts/t 。在量化V(Ri,ti)的Di时，在模型(1)中我们假设了fi的包到达速率服从泊松分布，在实际过程中很难知道一条流的包的λ强度，我们可以通过fi的包的速率来刻画。

在使用多重背包求解该问题时，可能存在饥饿分配的现象：某些低影响力的节点分配到0个Slot。未避免这种现象，我们预先从高影响力到低影响力的节点(Rs1-Rsk)为每一个节点分配1个Slot(在代价不超过总约束的情况下)，再使用多重背包求解，这样可以尽量避免饥饿问题。

多重背包求解时，除了存在饥饿问题外，求解效率是O(n^2)的复杂度。当T过大或者t过小时以及C较大时，求解最优解的时间时难以接受的。我们给出一种简单且高效的策略。算法2展示了这种方式。

时间序列

确定时间序列，在Model(1)中，体现的是各结点之间的重叠关系带来的影响，例如系统的采样的精度以及采样包的重复率。在第二章节中，我们使用高影响力私有化重叠流的方式近似解决了节点间重叠流所带来的精度的影响。因此，这一章节中，我们通过优化各节点Slot的采样序列，从而降低包的重复率。

该优化问题可以有如下公式定义。其中$$ 代表了两节点的时间Slot的重叠数量，$$代表ij两节点间的重叠流数量。因此，该公式体现了整个采样系统中流的重叠面积。

从该问题的定义上，可以使用搜索回溯的方式去求解最优解，但其是一个多项式时间内不可解的问题，因此我们考虑使用简单的贪心算法的方式来求解该问题的近似最优解。