

Flow-Awareness Maximization by Spatio-Temporal Sampling Collaboration in Software Defined Networks

He Cai¹, Jun Deng¹, Xiaofei Wang¹

¹Tianjin Key Laboratory of Advanced Networking, Tianjin University, Tianjin, China.

Abstract—With the proliferation of Internet applications and the explosion of traffic, Fine-grained flow-level information acquisition provides basic support for network management, TE, security analysis, and QoS. In traffic sampling, due to the scale of the network, the analysis capacity of the Collector, and the highly dynamic network, there are a large number of flows cannot be captured. In this paper, we focus on maximizing sampling accuracy and sample effective ratio, proposing a collaborative sampling model based on influence, which combines the three dimensional influences of the nodes to quantify the value of the nodes: the direct influence of the current active flow of nodes, the topological influence of the nodes, the historical influence of the nodes; And through the overlapping relationship of active flows between nodes, the cooperative sampling relationship of nodes at sampling timing is constructed. Based on this model, we propose the Agile Flow-Level Co-sampling strategy (AFCS), which solves the approximate optimal solution of CSBI through three steps: sampling point selection, sampling time allocation, and collaborative strategy optimization. We evaluated our strategy through a real large-scale topology, and the results show that it can effectively improve the sampling accuracy, especially in the micro-flow accuracy, and effectively improve the sampling efficiency.

I. INTRODUCTION

With the data traffic and network scale rapidly increasing, flow-level measurement is crucial to network management. In a large-scale network, accuracy is a vital statistic for flow-level sampling. High-accuracy sampling can provide more information for managers and network applications, such as intrusion detection systems (IDS), traffic engineering (TE) and application-aware networks. Especially for network security, over 80% of malware propagations via the internet can be prevented by capturing suspicious packets on the network and inspecting them by using security application solutions such as an IDS. Therefore, the higher the sampling accuracy, the higher the network security.

We consider two sampling approaches: the systematic packet sampling (SPS) and probabilistic packet sampling (PPS) [xx]. The former approach captures every packet for a sampling duration from a starting point in time. The other is a method to selectively capture packets from flows with a probability. They transmit packets to a unified analyzer (e.g., IDS) according to a certain routing rule. In a traditional network, there is no flexible way to deal with the changes of real-time traffic because of the closeness and isolation of network. Thanks to the birth of software-defined network,

SDN replaces the distributed, per-switch control planes of traditional networks with a (logically) centralized control plane that programs the forwarding behavior of all the switches in a given network. This centralized control plane, run on a controller that gathers traffic and other measurements from the network and uses the gathered information to compute and install forwarding behaviors in the switches. Recur to its centralized control and global visibility, there are more strategies for flow collection on the SDN. SDN controllers can gather traffic of any node with different strategies through a simple OpenFlow [2] protocol, which is incomparable in a traditional network.

When sampling in a large-scale network, in order to reduce the intrusion to the network, it is common to select several nodes to sample in the whole network. To maximize the sampling accuracy of flows, many studies have used social network theory to solve this problem [3,4]. And article [1,5] uses the number of active flows currently detected by the node as the influence to quantify the value of the node and select the sampling node. The sampling rate of nodes is another reason that affects the sampling accuracy. It is limited by constraints, such as the processing capacity of IDS, and is a limited sampling resource.

The Sampling rate of nodes is another reason that affects the sampling accuracy. It is limited by some constraints, such as the processing capacity of IDS, which is finite resources used for sampling. It is essential to determine a reasonable sampling rate for each node under a given constraint. For example, [3] uses Isometric allocation to determine the sampling rate for each node. Therefore, Selecting the sampling nodes and determining the sampling rate are two important factors that affect the accuracy.

The validity of sampling that represents the ratio of redundant samples to all sampled packets, is also a unignorable reason. Redundant packets are generated when a flow passes through multiple nodes at the same time, and at least two nodes capture the same packets from the flow. The higher the repetition rate, the more meaningless overheads of collector, which may reduce efficiency and even limit the promotion of accuracy. Therefore, our motivation is to reduce the repetition rate while determining the sampling node and sampling rate, so as to save sampling resources and ultimately maximize the sampling accuracy under the constraint of given collector

capacity.

In this paper, we use SPS method.1)When quantifying the node's integrated value, which can be used as a basis to determine the sampling node and sampling time,we consider not only the direct value brought by the current active flow of th node, but also the potential value consisting of the location in a topology and historical vitality of nodes in the sampling period. 2)We first proposed a cooperative sampling strategy among nodes, which considers not only node selection and time allocation, but also the sampling order of nodes in the time dimension for improving the accuracy. With the help of sampling strategy, the sampling order of each node is reasonably arranged in the time dimension,which can reduce the repetition rate markedly and further maintain the accuracy. 3)Therefore, we establish a model to quantify the maximum of flow-level sampling accuracy, in which the comprehensive impact of each node is used to evaluate the comprehensive value of the node, and the sampling order is used to optimize the sampling accuracy. 4) Based on the above model, we propose an AFCS strategy, which resolves the overall problem into three solvable sub-problems: node selection, time allocation, and cooperative sampling optimization. AFCS includes three related algorithms: XXX, XXXX, XXX. The output of the previous algorithm is the input to the next algorithm. And finally we obtains the approximate optimal solution of maximizing sampling accuracy through the heuristics method.

A. Related Work

B. Contributions

Our main contributions are summarized as follows:

- In the context of Flow-level traffic sampling, we first proposed the concept of cooperative sampling, and explained the effect of timing cooperation between nodes on sampling accuracy and sampling efficiency.
- For the first time, we modeled the problem of maximizing flow sampling accuracy. It constructs the cooperative relationship of the nodes through the overlapping relationship between the nodes, and quantifies the value of the nodes in the sampling period through the three dimensions of the node's current value, topological value and historical value.
- Based on this model, we propose an AFCS strategy for solving the approximate optimal solution of the model. The strategy consists of three algorithms: **Iterative comprehensive influence-based Top - K node selection based on flow privatization, Non-starved influence change-aware priority polling slot allocation, Co-Sampling slots ordering based on effectiveness maximizing greed.**And, We evaluated it through a real large-scale network topology.

II. SYSTEM DESCRIPTION AND PROBLEM STATEMENT

Figure 1 briefly demonstrates the process of node cooperative sampling with AFCS under two constraints: the collector capability $C_{packets}/T$ and the number of sampling nodes K . Before a new sampling period T begins, in order to maximize sampling accuracy over T , each node is assigned a different

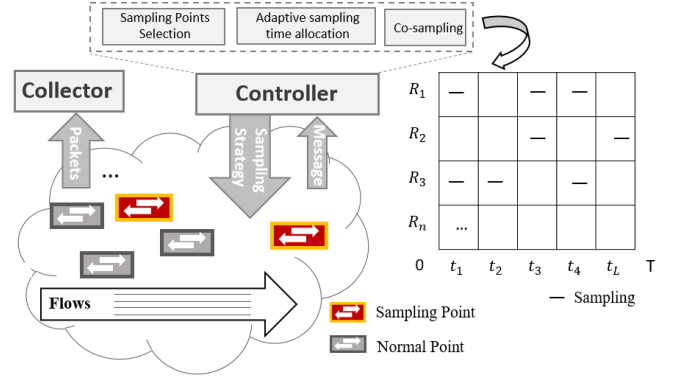


Fig. 1: Co-Sampling Architecture

number of sampling time slots and is arranged slot sequence at the same time. And all the sampling nodes work together to maximize the sampling accuracy of the flow.

A. Factors affecting sampling accuracy

In a large-scale network, the challenges to maximize the flow-level sampling accuracy are how to select K sampling nodes and how to allocate sampling time for these nodes over T without exceeding the constraints of C and K values. For node selection and time allocation, we should consider not only the current detected flow information, but also the possibility of new flow arriving at each node during a T cycle after running the sampling strategy. In addition, another important factor affecting the sampling accuracy is the repetition rate of the sampled packets, which represents the ratio of the repeated packets to all the packets. Collecting repeated packets not only occupies limited resources and reduces the sampling validity, but also reduces the accuracy. The higher the repetition rate, the more meaningless overheads of collector, which may limit the promotion of sampling accuracy or even reduce accuracy.

Repeated packets are generated when a data flow passes through multiple nodes at the same time, and at least two nodes capture the same packets from the flow. That is to say, *the overlap of the flows covered by nodes and the arrangement of sampling time sequence among nodes determine the repetition rate.* The overlapping relationship of the flows covered by the nodes and the overlapping relationship of the nodes in the sampling time dimension directly determine the repetition rate. To a certain extent, reducing unnecessary repetition rate is conducive to improving accuracy. From this perspective, in an optimal model, the overlap of flow among nodes, the order of nodes in the dimension of a cycle time, the value brought by the order of nodes, should be integrated into the node selection and time allocation. Since the goal of the system is to give priority to maximizing sampling accuracy, sometimes the repetition rate is unavoidable. And the repetition rate is necessary, if the value brought by it can maximize the sampling accuracy.

B. PROBLEM Model

Maximizing sampling accuracy of flows is the same as maximizing area coverage, as shown in Figure 3. The red solid area represents all flows passing through the whole network during the T , including currently active flows and subsequently arrive flows, denoted as F^c . The gray area is the set of active flows currently passing through the node, denoted as F_i^c , and the red area is the set of flows that the node may newly arrive in within T . When maximizing the area coverage, the gray area of the node and the red dotted area should be fully calculated. And the overlap of the areas between the nodes should also be considered, which is regarded as flow overlap mentioned above. For the value generated by the node, we should not only consider the value brought by its current flows (**Direct Value**), but also the value brought by the new arrival flows in T (**Potential Value**). The area coverage problem is like a traditional sampling solution, a two-dimensional problem: only consider selecting nodes and allocating sampling time. However, the effect of the overlap of flows between nodes on the sampling accuracy in the time dimension is not considered.

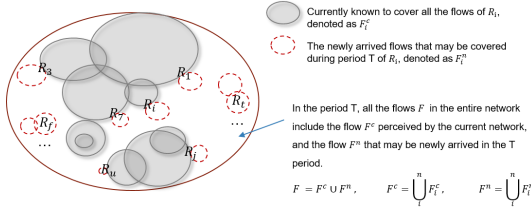


Fig. 2: Area coverage problem

In order to establish a model to maximize the sampling accuracy of flows, it is necessary to quantify the value and the cost of each node, based on the overlap of flows and the cooperation strategy in the time dimension. Let R as the set of all nodes. For any f_k in the set F_i^c , if any packet of f_k is captured by R_i , and then f_k is considered to be successfully captured by R_i . Assuming that the arrival of packets of a flow f_k obeys the poisson distribution with λ_k during the unit time t , then the expected number of current active flows captured by R_i is $\sum_{f_k \in F_i^c} P\{N_p^k(t) > 0\}$. Therefore, we can easily quantify the direct value that the node brings in t . But for the potential value of nodes, we do not know the number of flows that will arrive at each node during a T in the future. Even the arrival distribution of flows is known, we can not know the overlap of these flows among the nodes. Therefore, the potential value of nodes needs an independent quantification method. (independent to other nodes).

We propose a quantification approach based on comprehensive influence, which unifies direct value and potential value. Let t is the unit time, t is the smallest unit of sampling duration allocation, $t \leq T$. And let $L = T/t$ represents the number of time slots. The direct influence of any node R_i within a t can be quantified as (1). It is described as the expected number of currently active flows captured by the node within t divided by the total number of currently active

flows of whole network. The quantization method is based on the flow betweenness centrality (FBC) [2], which reflects the timely influence of nodes.

$$D_i = \sum_{f_k \in F^c} P\{N_p^k(t) > 0\} / |F^c| \quad (1)$$

And we evaluate the potential value of nodes from two dimensions: the power of the node in the network topology, and the proportion of the number of previous flows. We use standard betweenness centrality to measure the power of node R_i in the network [2], which we call the topology influence of nodes S_i and which represents the importance of the location in the network topology. The proportion of the number of previous flows reflects the activity of the node throughout the network lifecycle, called history influence H_i . $H_i = TF_i / TF$, where TF_i represents the number of flows that has passed R_i so far, and TF is the total number of flows that have passed the network so far.

Therefore, the potential value of the node during a T can be quantified using S_i and H_i . The higher the S_i and H_i , the higher their potential value and the more likely they are to go through more flows in the future. And during a t , the compositive value generated by the node can be expressed as $\alpha \cdot D_i + \beta \cdot \frac{S_i}{L} + \gamma \cdot \frac{H_i}{L}$, where $\alpha + \beta + \gamma = 1$ represents the weighted sum of the three influences.

The cost of R_i is the total number of packets collected in unit time. We use w_i to represent the total number of packets detected by R_i within t . w_i can be expressed as (2), where v_i is the current packet rate (packets/ T) of R_i . Because of the subsequent arrival of unknown traffic, we use the ratio of potential value to direct value to infer w_i .

$$w_i = \frac{1}{L} \cdot [v_i + v_i \cdot \frac{\beta \cdot \frac{S_i}{L} + \gamma \cdot \frac{H_i}{L}}{\alpha \cdot D_i}] \quad (2)$$

We give the Influence-based maximization flow sampling accuracy collaborative sampling model in (3), where s^l ($l=1, 2, \dots, L$) represents the different t slots in T . (4) (5) describe the constraints that the model should satisfy.

$$\max \sum_i^n (\alpha \cdot \delta(D_i, |\tilde{S}_i|) + \frac{|\tilde{S}_i|}{L} \cdot (\beta \cdot S_i + \gamma \cdot H_i)) - IR \quad (3)$$

$$\text{subject to: } \sum_i^n f(\tilde{S}_i) \leq K, f(\tilde{S}_i) = \begin{cases} 1, & |\tilde{S}_i| \geq 1 \\ 0, & |\tilde{S}_i| = 0 \end{cases} \quad (4)$$

$$\sum_i^n w_i \cdot |\tilde{S}_i| \leq C \quad (5)$$

The goal of the model is to allocate the corresponding sampling time for each R_i under the constraints of given C and K , and to determine the sampling cooperation strategy among nodes, when maximizing the influence of the system in the period T or maximizing the flow-level sampling accuracy. \tilde{S}_i represents the set of slots allocated to R_i . The different slots in \tilde{S}_i represent the sampling arrangement of the node in the time dimension. The larger the size $|\tilde{S}_i|$, the greater the comprehensive influence (value) generated by the node. The

relationship between the comprehensive influence of nodes and $|\tilde{S}_i|$ is given in (6).

$$\delta(v_i, |\tilde{S}_i|) = \begin{cases} v_i \cdot |\tilde{S}_i|, & v_i \cdot |\tilde{S}_i| < |F_i^c| \\ |F_i^c|, & \text{ELSE} \end{cases} \quad (6)$$

When $D_i \cdot |\tilde{S}_i| = |F_i^c|$, it means that R_i can capture all the flows of F_i^c under $|\tilde{S}_i|$. Then its direct value will not continue to increase with the increase of $|\tilde{S}_i|$, but the potential value will continue to increase. Therefore, the value generated by a single slot of a node is related to the number of slots the node already has, not independent.

In (3), the first part represents the sum of the combined values of all nodes under their respective \tilde{S}_i ; And the subtraction part IR indicates the sum of the direct influences of the resampled flows at all the same time slots which described in (7), where the $\psi(f_k, s^l)$ calculates the number of times each flow is resampled at slot s^l . $U = \{R_i, f_k \in F_i^c \wedge s^l \in \tilde{S}_i\}$ is a set of routers which contain f_k and sampling at time s^l . As mentioned above, the packet repetition rate is generated by more than one nodes sampling the same flows at the same time. Therefore, the model takes the overlapping relationship of flows between nodes and the arrangement of nodes in the sampling time dimension as one of the factors that influence the accuracy of flow sampling. In order to maximize (1), *each sampling node must have a reasonable slots arrangement for sampling, collaboration of sampling nodes in time dimension, in exchange for the improvement of sampling accuracy.*

$$IR = \frac{\alpha}{|F^c|} \cdot \sum_{f_k \in F^c} \sum_{l=1}^L (P\{N_p^k(t) > 0\} \cdot \psi(f_k, s^l)) \quad (7)$$

$$\psi(f_k, s^l) = \begin{cases} |U| - 1, & |U| \geq 1 \\ 0, & |U| = 0 \end{cases} \quad (8)$$

We explained the quantization process of the model and illustrated it as Figure3. The goal of the model is to allocate reasonable \tilde{S}_i sets for all R_i under constraint conditions. As figure3 shows, when the system gets the optimal solution of $S^1..S^n$, it not only reflects the optimal node selection, the optimal time allocation, but also reflects the optimal slot order arrangement, which is determined by the cooperation strategy between nodes. At the same time, each node actually avoid overlapping on the same slot as much as possible, so that the efficiency of the sampling system is improved. The time complexity for solving this model is $O(2^{|R| \cdot L})$.

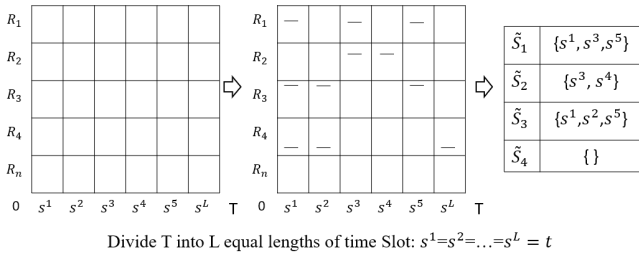


Fig. 3: Overview of model

III. AGILE FLOW-LEVEL CO-SAMPLING STRATEGY(AFCS)

Since the above model is complicated, we divide it into three sub-problems: K node selection, Slot allocation, and collaborative sampling optimization. The solution to each sub-question will be the input to the next sub-question. Through three steps, the approximate optimal solution of the total problem is obtained. In this section, we detail each sub-question and propose the corresponding algorithms.

A. Iterative comprehensive influence-based Top – K node selection based on flow privatization

In the selection of K sampling points, we only focus how to use the static properties of the nodes to quantify their comprehensive influence, which is independent of time allocation. In (3), when the number of slots and the cooperative relationship between nodes (slots order) are not considered, the optimization formula can be expressed as (9) and the constraint is (10); The goal of the optimization model (9) is to select K sampling nodes without considering the influence of time on the comprehensive influence of the nodes, so as to maximize the comprehensive influence of the system. Similar to the model of (3), the former part sums the comprehensive influence of the selected nodes, and the latter part subtracts the influence part of the repeated calculations.

$$\max[\sum_{i=1}^n (\alpha \cdot D_i + \beta \cdot S_i + \gamma \cdot H_i) \cdot \varphi(i) - \alpha \cdot \sum_{f_k \in F^c} \hat{\psi}(f_k)] \quad (9)$$

$$\text{subject to: } \sum_{i=1}^n \varphi(i) = K \quad (10)$$

In(9), $D_i = |F_i^c|/|F^c|$, and $\varphi(i)$ is 0 means that R_i is not selected, 1 means R_i is selected; $\hat{U} = \{R_i, f_k \in F_i^c \wedge \varphi(i) = 1\}$ indicates how many selected nodes cover f_k , so the (11) indicates how many selected nodes are repeatedly calculated f_k as their direct influence. Therefore, the subtracted part expresses that any flow will only be calculated once as a direct influence, in other words, any flow will only be calculated as direct influence by only one of the selected nodes, *the flow is privatized by a selected node during the selection process.*

$$\hat{\psi}(f_k) = \begin{cases} |\hat{U}| - 1, & |\hat{U}| \geq 1 \\ 0, & |\hat{U}| = 0 \end{cases} \quad (11)$$

Therefore, it is possible to iteratively calculate the top K nodes with the most comprehensive influence, the highest comprehensive influence in each round will be selected, and privatize all the flows it covers (but not including the flows privatized by other selected nodes in the previous); That is, for the direct influence of the candidate R_i in the k round, use the flow set \tilde{F}_i^{cs} to calculate the direct influence; Let F_k^{cs} denote the set of flows privatized by the selected node in the k_{th} round, so $\tilde{F}_i^{cs} = F_i^c - \bigcup_{m=1}^{k-1} F_m^{cs}$, and the subtracted portion indicates that it has been privatized by the selected node of the previous k-1 round; If R_i is selected in this round, the flows in \tilde{F}_i^{cs} is R_i exclusive, then \tilde{F}_i^{cs} written as F_k^{cs}, R_i

is written as R_k^s ; In the $k + 1$ round selection, even if the other candidate nodes cover these flows, these flows have been privatized by the R_i , will not bring value to these candidate nodes in the $k + 1$ round. The K-round selection is carried out, and the nodes with the highest comprehensive influence are selected in each round, through the process of privatizing the flows with the most influential nodes in each round, any two selected nodes are: $F_i^{cs} \cap F_j^{cs} = \emptyset$, and the comprehensive influence of each node is the highest in the corresponding round, so the K selected nodes guarantee (9) to be maximized under the constraint (10). so the K nodes are the optimal solutions of (9). D_i^k indicates the direct influence of the candidate node R_i in the k_{th} round selection, which $D_i^k = |F_i^{cs}|/|F^c|$. (12) describes the iterative calculation formula for the comprehensive influence of the candidate nodes for each round of selection.

$$I_i^k = \alpha \cdot D_i^k + \beta \cdot S_i + \gamma \cdot H_i \quad (12)$$

Algorithm 1 gives the solution process, and the time complexity of the algorithm is $O(K * |R|)$. Finally, the K node set R^s and the F^{cs} set corresponding to each node can be obtained.

Algorithm 1 Sampling Point Selection

```

1: define  $R^s = \emptyset$  // The Set of Selected Routers
2: for  $k = 1; k < K; k++$  do
3:   for each  $R_i \in R - R^s$  do
4:     if  $I_i^k > \max$  then
5:        $\max = I_i^k$ 
6:        $R_k^s = R_i$ 
7:   put  $R_k^s$  into  $R^s$ 
8:    $F_K^{cs} = F_i^c - \bigcup_{m=1}^{k-1} F_m^{cs}$ 
9: return  $R^s$ 

```

B. Non-starved influence change-aware priority polling slot allocation

After the nodes is selected, the number of slots needs to be allocated to these nodes when the constraint (5) is satisfied and the slots sequence between nodes is not considered. The optimization model of the subproblem is (13), which is still judged by formula (6). The goal of the model is to maximize system value under the given R^s and constraints (5).

$$\max[\alpha \cdot \delta(D_i, CNT[i]) + \frac{CNT[i]}{L}(\beta \cdot S_i + \gamma \cdot H_i)] \quad (13)$$

The reason why the model does not need to consider the influence of overlapping flows between nodes on the total influence of the system is that we use the privatized flows set of each node F^{cs} to quantify the direct influence of each node, and arbitrarily $F_i^{cs} \cap F_j^{cs} = \emptyset$, so the direct influence in the node R_i in unit time t is (14).

$$D_i = \frac{\sum_{f_k \in F_i^{cs}} P\{N_p^k(t) > 0\}}{|F^c|} \quad (14)$$

For the cost w_i of the node in unit time t , still uses F_i^c to calculate. This is because we only logically eliminate the

overlapping flows between nodes, but there is still no change in the overall rate of the underlying nodes. $CNT[1..K]$ represents the number of slots for each node.

In the sub-problem optimization model, the comprehensive influence of the nodes increases with the increase of the number of Slots. When the situation of ELSE in formula (6) is satisfied, another growth trend will be presented, and the trend depends on S_i and H_i . We have described the reasons in the construction of the model of (ref1). That is to say, the value generated by a single Slot is related to the number of Slots, not independent, so it is not possible to solve the optimal solution with multiple backpacks.

We give a simple and efficient allocation algorithm to achieve: the simple influence of the node's comprehensive influence per unit time t from high to low polling allocation, so that the current high-influence nodes preferentially allocate slot. After each round of allocation, if a node satisfies the ELSE condition of formula (6), the comprehensive influence of the node in unit time t is $\beta \cdot S_i + \gamma \cdot H_i$, which does not satisfy the ELSE condition in (6), continue to use the comprehensive influence of the three weighted sums. The algorithm stops until $C = 0$ or C cannot be allocated. Algorithm 2 gives a description of the process. The algorithm perceives the change in the comprehensive influence per t due to the change in the number of node slots. Each round guarantees that the current high-influenced nodes preferentially allocate slot, and the polling allocation method can also avoid the starvation of some low-influence nodes. $CNT[1..K]$ can be obtained. The time complexity of the algorithm is $O(\frac{C}{\min\{w_i\}})$.

Algorithm 2 Impact Priority Polling Allocation Slots

```

1: Define  $CNT[1..K] = 0; \tilde{I}[1..K] = 0;$ 
2: while  $C > 0$  OR  $C$  is different from the last round do
3:   for each  $R_i^s \in R^s$  do
4:     if  $R_i$  satisfy  $\delta(D_i, CNT[i])$  the ELSE condition then
5:        $\tilde{I}[i] = \frac{1}{T/t} \cdot (\beta \cdot S_i + \gamma \cdot H_i)$ 
6:     else
7:        $\tilde{I}[i] = \alpha \cdot \frac{v_i \cdot 1}{|F^c|} + \frac{1}{T/t} \cdot (\beta \cdot S_i + \gamma \cdot H_i)$ 
8:   Descending sorting  $\tilde{I}$ ;
9:   for  $i = 1; i < K; i++$  do
10:    if  $C \geq w_i$  then
11:       $CNT[i]++$ 
12:       $C = C - w_i$ 
13: return  $CNT[1..K]$ 

```

C. Co-Sampling slots ordering based on effectiveness maximizing greed

In model (3), we have described the effect of cooperative sampling between nodes on sampling accuracy and the effective ratio of sampling. As shown in Fig.(3), the cooperation in time dimension between nodes is reflected in the order of sampling Slots of each node. Two nodes covering same flows should be avoided sampling in the same slot. They

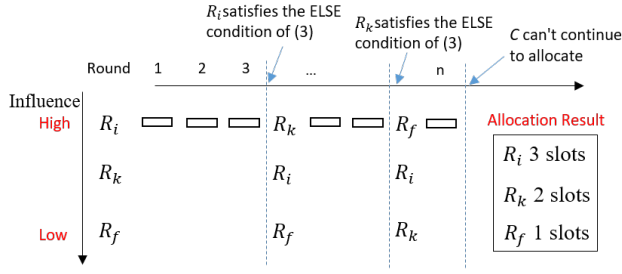


Fig. 4: Slot allocation example

should be arranged with the appropriate sampling order and cooperate with each other to maximize the effectiveness of the entire sampling system during period T. For this sub-problem, formula (15) is its optimization model. The optimization goal of the model is to minimize the number of resampled flows under given the number of Slots of each node. For any two sampling points R_i, R_j , assuming that S_i, S_j are their Slot sets respectively, then $|S_i \cap S_j|$ indicates the number of times they are sampled under the same Slot; $|F_i^c \cap F_j^c|$ indicates the number of flows that they cover the same. Therefore, the number of flows that are resampled between two nodes can be expressed as: $|S_i \cap S_j| \cdot |F_i^c \cap F_j^c|$. How to reasonably arrange the sampling time slot sequence of each node in the period T, so that the number of resampled flows of the whole system is minimized, thereby ensuring the minimum sampling packet repetition rate, so that the effectiveness of the whole sampling system is maximized.

$$\min \sum (|\tilde{S}_i \cap \tilde{S}_j| \cdot |\tilde{F}_i \cap \tilde{F}_j|), \forall i, j \wedge j > i \quad (15)$$

This problem can be solved by using the search backtrack method, but it is a problem that cannot be solved in a polynomial time. Therefore, we consider a simple greedy algorithm to solve the approximate optimal solution of the problem. Algorithm 3 gives a description of the process. The CNT array has been calculated in the previous section, indicating the number of slots for each node. At the beginning of the algorithm, initialize the M^{slot} two-dimensional array to store the placement relationship between R_i and s^l : $M^{slot}[i][l] = 1$, which represents R_i node sampling at s^l . In each round, an idle slot is selected for all nodes of $\forall i, CNT[i] \neq 0$, and the selected slot makes the number of resampled flows of the whole system the least compared to other optional slots. Through each round, the nodes greedily chooses a slot that minimizes the current the number of resampled flows of the entire system, when $i, CNT[0] = 0$, the slots order of each node are selected, an approximate optimal solution is obtained. The time complexity of the algorithm is $O(L^2 \cdot K)$. Fig. 6 demonstrates the process. In the example, the approximate solution we solved by this algorithm is 6 and the optimal solution is 5.

IV. EXPERIMENTS AND RESULTS

In order to verify the effectiveness and performance of our algorithm, we have built a laboratory bed based on floodlight

Algorithm 3 Order of Time Slot Based on Greedy

Input: M, S, c_j

- 1: **while** $CNT[i] > 0, \exists i \wedge i = 1, 2, \dots, k$ **do**
- 2: **for** $i = 1; i \leq K; i++$ **do**
- 3: **if** $CNT[i] > 0$ **then**
- 4: $Min = \text{Max Integer}$
- 5: **for** $l = 1; l < \frac{T}{t}; l++$ **do**
- 6: **if** $M^{slot}[i][l] = 0$ **then**
- 7: $temp = \sum_{j=1 \wedge j \neq i}^K (|\tilde{F}_i \cap \tilde{F}_j| \cdot M^{slot}[j][l]) + H[l]$
- 8: **if** $temp < Min$ **then**
- 9: $Min = temp; Sp = l$
- 10: $M^{slot}[i][Sp] = 1; CNT[i]--; H[Sp] = Min$
- 11: **return** M^{slot}

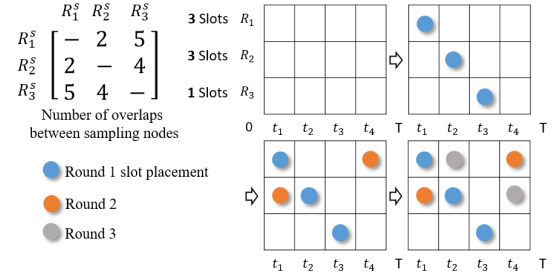


Fig. 5: Illustrating sampling node slots placement based on greedy algorithm

controller and openswitch + mininet. The whole experimental bed contains 12 Dell XPS hosts, 20 core CPU, and Ubuntu 16.06.2 LTS. One runs a floodlight controller that fuses our algorithm, the other runs a data collector, and the remaining 10 deploy a network topology with 110 switch nodes and 50 host nodes. The experimental traffic dataset comes from the open project "the WIDE Project". We selected data from 14:00-14:15 in August 6, 2018. After cleaning and screening, we collate 5000 data streams for experiments. In the experiment, the number of data streams changed from 1000 to 5000. We implement four algorithms: Random-K, top-K based on the extended median centrality, top-K based on the standard median centrality, our algorithm XXX. Based on the above four algorithms, we have made a comparative experiment in three measurement mechanisms: sampling accuracy, packet repetition rate and the number of rat streams collected. Fig. x shows the comparison of sampling accuracy in different algorithms. Our algorithm is 7% higher than Top-k and over 20% than the other two algorithms. From Fig. x, we can see that different algorithms do almost the same amount of elephant flow collection. In fact, our algorithm only takes more part of the rat flow than other algorithms. And Fig. x shows that our algorithm is effective in reducing duplication and reducing it by more than 30%.

V. CONCLUSION

- Lab environment

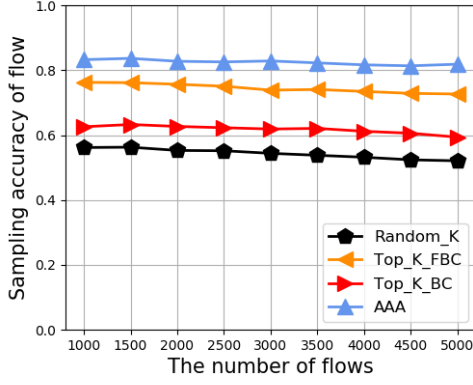


Fig. 6: accuracy comparison with respect to different algorithms

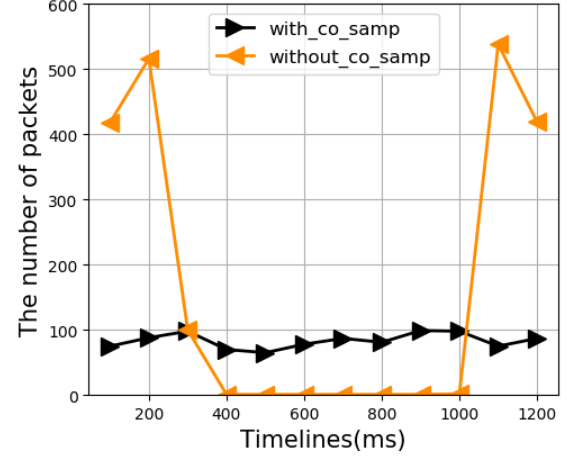


Fig. 9: comparison of sampling flow number at different slot

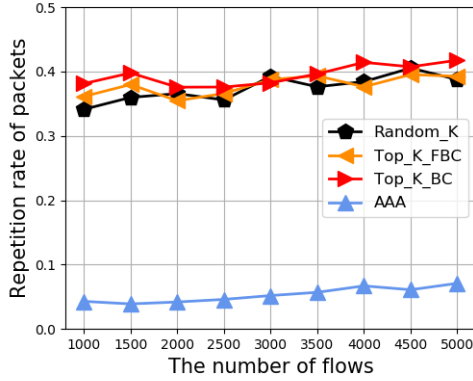


Fig. 7: repetition rate comparison with respect to different algorithms

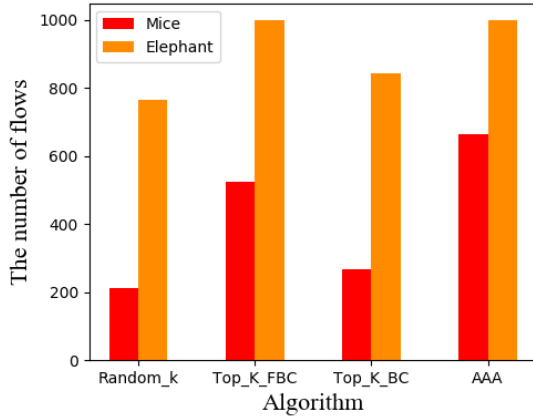


Fig. 8: comparison of different algorithms in number of mice flow

- Sampling accuracy comparison
- Sampling repetition rate comparison
- Greedy centrality algorithm experimental results

- Deduplication rate algorithm comparison
- Experimental comparison of adaptive co-sampling algorithm

REFERENCES

- [1] S. Yoon, T. Ha, S. Kim and H. Lim, Scalable Traffic Sampling using Centrality Measure on Software-Defined Networks, in *IEEE Communications Magazine*, pp.43-49, July 2017.
- [2] M. Malboubi, L. Wang, C.N. Chuah, P. Sharma, Intelligent SDN based Traffic (de)Aggregation and Measurement Paradigm (iSTAMP), in *IEEE INFOCOM*, Apr 2014.
- [3] L. Tong and W. Gao, Application-Aware Traffic Scheduling for Workload Offloading in Mobile Clouds, in *IEEE INFOCOM*, pp.1-9, Apr 2016.
- [4] J. Jiang, S. Ma, B. Li and B. Li, Symbiosis: Network-Aware Task Scheduling in Data-Parallel Frameworks, in *IEEE INFOCOM*, pp.10-14, Apr 2016.
- [5] P. Bakopoulos, K. Christodouloupolos, G. Landi et al, NEPHELE: An End-to-End Scalable and Dynamically Reconfigurable Optical Architecture for Application-Aware SDN Cloud Data Centers, in *IEEE Communications Magazine*, pp.178-188, Feb 2018.
- [6] J. Xu, J.Y. Wang, Q. Qi, H.F. Sun and B. He, IARA: An Intelligent Application-aware VNF for Network Resource Allocation with Deep Learning, in *IEEE SECON*, pp.1-3, June 2018.
- [7] J. Suh, T.T. Kwon, C. Dixon, W. Felter and J. Carter, OpenSample: A Low-latency, Sampling-based Measurement Platform for Commodity SDN, in *IEEE ICDCS*, pp.228-237, July 2014.
- [8] Z. Su, T. Wang, Y. Xia and M. Hamdi, CeMon: A Cost-effective Flow Monitoring System in Software Defined Networks, in *Computer Networks*, pp.101-115, Dec 2015.
- [9] N.F. Huang, C.C. Li, C.H. Li, C.C. Chen, C.H. C and I.H. Hsu, Application Identification System or SDN QoS based on Machine Learning and DNS Responses, in *APNOMS*, pp.407-410, Sept 2017.
- [10] S. Jeong, D. Lee, J. Hyun, J. Li, and J.W. Hong, Application-aware Traffic Engineering in Software-Defined Network, in *APNOMS*, pp.315-318, Sept 2017.
- [11] G. Cheng and Y. Tang, eOpenFlow: Software Defined Sampling via a Highly Adoptable OpenFlow Extension, in *IEEE ICC*, pp.1-6, May 2017.
- [12] S. Zhao and D. Medhi, Application Performance Optimization Using Application-Aware Networking, in *IEEE NOMS*, pp.1-6, Apr 2018.
- [13] M. Malboubi, S.M. Peng, P. Sharma and C.N. Chuah, A Learning-based Measurement Framework for Traffic Matrix Inference in Software Defined Networks, in *Computers & Electrical Engineering*, Dec 2017.
- [14] K. Bilal, S.U. Khan, L. Zhang et al, Quantitative comparisons of the state-of-the-art data center architectures, in *Concurrency & Computation Practice & Experience*, Dec 2017.