AAA: High Agile Adaptive Flow-Awareness Network for SDN

He Cai[1], Jun Deng[1], Xiaofei Wang[1]
[1]Tianjin Key Laboratory of Advanced Networking, Tianjin University, Tianjin, China.

***Abstract*—With the proliferation of Internet applications and the explosion of traffic, Fine-gaine's 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 capcity of the Collector,and the highly random and dynamic network environment, there are a large number of uncapable flows. In this paper, we focus on maximizing the sampling accuracy, and proposing the Influence Maximization Model(IMM) from the three dimensions of sampling node selection, time allocation and collaboration between nodes.Based on the optimization model, we propose three heuristic algorithms to solve the approximate optimal solution. We implemented the AAA platform and evaluated the performance of the algorithms using real network topology.**

1. Introduction

With the data traffic and network scale rapidly increaing, there exists huge demand for scalable network management. Meanwhile, network monitoring and application awareness play a increasingly critical role in Quality of Service(Qos),Traffic Engineering(TE) and cyber security. Briefly, application awareness is a basic technology to enhance automation and intelligence of the network. It is divided into two processes: packet acquisition and traffic identification. Packet acquisition refers to capturing packets from switches through a mechanism or an algorithm. Traffic identification refers to parsing the five-tuple information of packets from different layer according to OSI model ,then recognizing the application layer protocol with the help of DPI tools. Application-aware network can improve the visibility of itself , promote integration of different business and eliminate faults quickly .However, the application awareness need integrate the high precision,high efficiency with real time, which is still a challange owing to the volume and variety of data in the large-scale network.

Software-defined network (SDN) is a new technical architecture which decouple the network control plane from the data-forwarding plane. It advocates building an open and programmable network to provide flexible, central controlled(or centralized) and globally visible network services, through which SDN can facilitate the operation and maintenance of the data center(DC) network. In a software-defined network, packet acquisition depends on OpenFlow(OF) protocol,which is varied from the Netflow and Sflow used in traditional networks.

Based on port, payload, and traffic behavior characteristics, DPI can identify a variety of information including the application layer protocol of a data flow, and be applied in application-aware network. In traditional networks, DPI devices are bound to the data plane, which makes it impossible to visualize global fine-grained traffic in real time. Therefore,many people are concerned about the research and optimization of the combination of SDN and application awareness. However, most of the current solutions are to deploy DPI in the SDN controller.In this case, parsing each single packet will be computationally heavy for controller. In addition, network scale, number of sampling nodes, sampling frequency and repetition rate of packet all increase performance consumption of controllers.On the other hand, in order to improve the accuracy of application recognition, the system must be able to capture continuous packets of the same flow regarding to the characteristics of DPI. To solve the above problems, a agile, adaptive and cooperative sampling mechanism which can be applied to large-scale data center network is urgently needed.

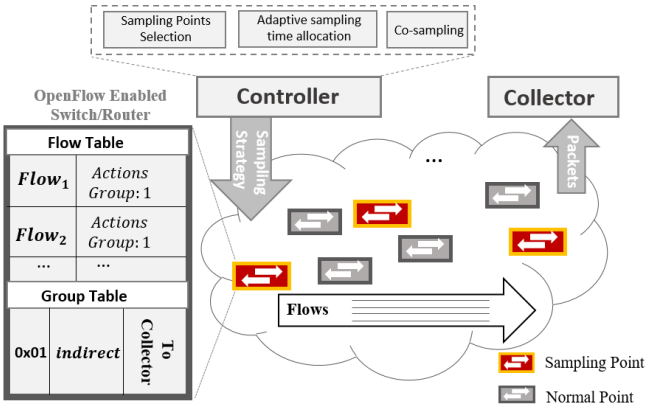


Figure 1: The System Architecture Of AAA

2. Related Work

Our main contributions are summarized as follows:

- In the context of Flow-level Sampling, we focus on maximizing the sampling accuracy. From the three dimensions of node selection,sampling node cooperation and time allocation, the IMM impact maximization model is constructed. We first explained the effect of cooperative

sampling between nodes and packet repetition rate on sampling accuracy.
- Based on the IMM optimization model, we split it into three sub-problems: sampling node election, Slot time allocation, and cooperation between nodes, and three heuristic algorithms are proposed to solve the approximate optimal solution of IMM.
- We built the AAA platform and evaluated the performance of the algorithms using a real large-scale WAN topology. Finally, an application identification demo is implemented: App-Awareness.

3. SYSTEM DESCRIPTION AND PROBLEM STATEMENT

Figure 1 shows the framework of AAA. In the AAA mode, the SPS mode is used to collect the flow table entries of each group. All packets representing all the flows passing through the switch are copied to the unified group table entry to perform the actions defined by the group entry. When the controller controls the initialization of the group of entries, the action is initialized to point to the collector or discarded. Therefore, when the controller needs to control a switch to sample or stop sampling, it only needs to simply send a Group Mod message to the corresponding switch. When the action is Drop, the sampling is stopped. When it is directed to the exit of the collector, the sampling is started. . This not only makes use of the pure OpenFlow protocol, but also controls it more precisely based on the controller's global. Assuming that the sampling period is T, through the adaptive coordination algorithm of AAA, the sampling points are selected and the sampling duration is allocated for them, and the sampling order is determined and the strategy is decentralized to the corresponding switch to make them cooperate sampling.

Sampling in large-scale networks, while satisfying the low intrusion of the network and the limitation of the Collector (IDS e.g) analysis capability, maximizing the sampling accuracy of the Flow-level and increasing the sampling efficiency ratio is a huge challenge.Therefore, highly agile and adaptive algorithms are needed to set the sampling strategy based on the real-time conditions of the current network. When the network size is large, assuming that the number of switches is n, $K(K <= n)$ nodes are generally selected for sampling, and K is determined according to actual conditions.In each sampling period T, if the capacity of the collector is C $packets/T$; Within T, the total number of sampling packets must be less than or equal to C.For flow-level sampling, under the given C and K constraints, the reasons for affecting the sampling accuracy include the selection of sampling nodes and the allocation of sampling time for each node.Another reason should be the sampling effective ratio (sampled number of non-repetitive packets/number of total sampling packets).The collection of repeated packets not only occupies a limited sampling resource, but also limits the sampling accuracy, and also reduces the effectiveness of the upper application (IDS e.g).The repeated packets is generated because multiple switching nodes cover same flows and are sampled at the same time. Therefore, *each sampling node should consider the overlapping relationship of the flows, and have reasonable coordination in the respective sampling time schedules*, so that the overlapping time of the system can be as small as possible, thereby reducing the repetition rate and improving the sampling accuracy.

3.1. IMM Model

The most reasonable node selection and time allocation, the optimal co-sampling strategy to maximize the flow-level sampling accuracy. Therefore, we consider from these three perspectives to model the maximum flow-level sampling accuracy. First we propose an intuitive perspective based on maximizing area coverage to analyze the problem. Figure 2 shows this idea: In a sampling period T, the set of flows F in the entire network is like an area depicted by a solid red line. R1..Rn, each node is covered by 0 or more streams, the set F_i^c, which is a shaded area in gray in Fig. 2, which represents the value of the node for the coverage of the known stream in the whole network. recorded as dynamic value. The red dashed area, which is the newly arrived stream that node Ri may cover in T, represents the potential value of the node in T time: these streams arrive after the sampling strategy is decentralized and are therefore not perceived by the sampling algorithm . The overlapping area between the gray areas is the overlapping part of the flow covered between the nodes, $F_i^c \bigcap F_j^c$. The overlapping portion between the red dotted areas is the overlapping portion of the newly arrived flow covered between the nodes. Some nodes contain both gray and red areas, The greater the number of streams detected by the node, the greater the value of the node. Therefore, the Flow-Level sampling problem can be intuitively converted into an area coverage maximization problem. That is, under the given collector processing power and other constraints, the sampling time allocation of each node is realized, so that the coverage area is maximized (the maximum number of stream coverage).

On the problem of maximizing area coverage, for the same stream, if there is temporal overlap in sampling on different nodes, the overlap should be subtracted when calculating the overall coverage area. However, the red region of each node is unknown to the sampling strategy. Even if the distribution model of the flow can be analyzed, the overlapping relationship of these unknown flows at each node cannot be known. Therefore the potential value of a node requires an independent (independent of other nodes) quantization. For a node, its intermediateity in the Topology [2] and the proportion of its historical flow can be used to evaluate the potential impact of the node, and these influences represent the potential value of the node, namely: potential impact The greater the force, the greater the value that the node may create in a unit of time. The value generated by the node during unit time t should be equal to the dynamic value + potential value. Assume that the arrival of the packet of the stream f_i obeys the Poisson

distribution of da, so if any switch captures at least one packet of f_i, then the stream is considered to be successfully captured, f_i The probability of being captured in unit time t is $PN_p^k(t) > 0$. For R_i, if it is assigned a t, the dynamic value brought by the node (actually the number of expected to capture the known stream) is , then its area coverage is $D_i/|F^c|$, the quantization method is based on an extended version of [2] The mediation center metric, which we call the dynamic impact of R_i over the t-sampling duration. The OSPF-based mediation in Topology measures the influence of nodes in the topology. We use a standardized intermediate degree [2], and the influence of R_i in the topology over the entire sampling period T is recorded as S_i. If the higher the S_i, the node is likely to go through more streams. If at some point in the network, the dynamic influence of R_i is greater than R_j, that is, the current R_i has more streams, but the dynamic impact of R_j is higher than R_i, so in the following time, R_j has a greater probability of going through more streams. The historical flow ratio of node R_i reflects the activity of the node throughout the network life cycle. We call it the historical influence of R_i over the entire sampling period T, denoted as H_i, $H_i = TF_i/TF$. The potential influence of R_i in period T is the combination of S_i and H_i. The combined influence of R_i on the sampling time per unit time t can be quantified as $(t <= T)$, which is the weighted sum of aby.

Therefore, we present the Impact Maximization Quantization Model (IMM) formula (1), which uses the influence maximization approach to maximize the sampling of the stream. Since the time is continuous, first let t be the unit time length, where $t <= T, letl = T/t$, which means that there are 1 unit Slot in the T period, and each slot is recorded as s1...sl. Where $\overset{S}{i}$ represents the set of Slots allocated for R_i. When $|S|$ is larger, the comprehensive influence of the nodes is larger, but (3) gives the comprehensive influence of the nodes with Slot. The number of judgment conditions is increased. When $v_i * |^S| > |F_i^c|$ is: the node realizes full capture of the current stream passing through it, then more Slot allocation will not continue to enhance its dynamic influence, and will only continue to enhance the potential impact. force. Therefore, the combined influence of all nodes is summed, and the dynamic influence of the overlapping part of the whole system is subtracted to obtain the total influence of the system.

$$\max \sum_i^n (\alpha \cdot \frac{\delta(v_i, \left|\widetilde{S}_i\right|)}{|F^c|} + \frac{\left|\widetilde{S}_i\right|}{T/t} \cdot (\beta \cdot S_i + \gamma \cdot H_i)) \\ -\frac{\alpha}{|F^c|} \cdot \sum_{f_k \in F^c} \sum_{l=1}^{T/t} \left(P\left\{N_p^k(t) > 0\right\} \cdot \psi\left(f_k, s^l\right)\right) \tag{1}$$

subject to:

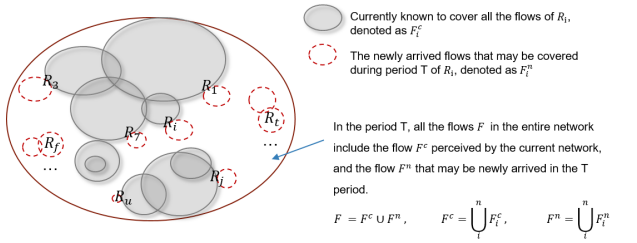
$$\delta\left(v_i, \left|\widetilde{S}_i\right|\right) = \begin{cases} v_i \cdot \left|\widetilde{S}_i\right|, v_i \cdot \left|\widetilde{S}_i\right| < |F_i^c| \\ |F_i^c| \quad , \text{ ELSE} \end{cases} \tag{2}$$

$$U = \left\{R_i, f_k \in F_i^c \wedge s^l \in \widetilde{S}_i\right\} \tag{3}$$

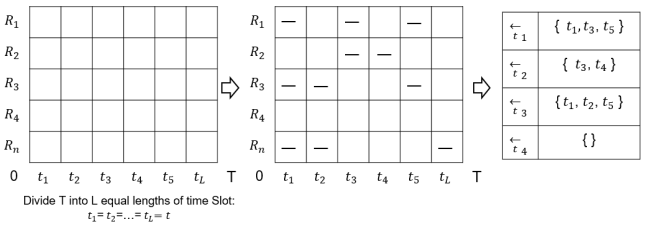
$$\psi\left(f_k, s^l\right) = \begin{cases} |U| - 1, |U| \geq 1 \\ 0 \quad , |U| = 0 \end{cases} \tag{4}$$

$$\sum_i^n f(\widetilde{S}_i) \leq K, \quad f(\widetilde{S}_i) = \begin{cases} 1, & \left|\widetilde{S}_i\right| \geq 1 \\ 0, & \left|\widetilde{S}_i\right| = 0 \end{cases} \tag{5}$$

$$\sum_i^n w_i \cdot \varphi(\widetilde{S}_i, s^l) \leq C \cdot t, \forall s^l \wedge \varphi(\widetilde{S}_i, s^l) = \begin{cases} 0, s^l \notin \widetilde{S}_i \\ t, s^l \in \widetilde{S}_i \end{cases} \tag{6}$$



(a) area coverage



(b) time slot allocation

Figure 2: overview of model

For the model of maximizing impact problems, the coverage of known traffic, the coverage of unknown new flows, the selection of nodes, the allocation of time slots, and the scheduling of Slot time series are fully considered. In the scheduling of time Slot, because the overlapping flows between nodes will cause conflicts under the same Slot, the overall coverage is reduced, so in order to get the optimal solution, the more overlapping the two nodes, if they are both When more than 0 Slots are allocated, the number of Slots they overlap should be as small as possible, which in turn solves another important sampling problem: how to reduce the problem of duplicate packets. Because each node contains the same stream, in order to optimize the results, it will be mutually exclusive on the Slot. Therefore, in the optimized solution, the sequence of $\widetilde{S}_i$ of each node can make the area covered by overlapping sampling in the whole cycle. Minimized, thus greatly reducing the repetition rate of the package while ensuring maximum sampling accuracy. Therefore, the nodes are cooperative, and through the constraint relationship between each other, the flow coverage relationship, the static topology relationship, the activity degree, etc., and finally maximize the number of sampled streams. (In the sampling process, when multiple nodes are

sampled, a large number of duplicate packets are generated, because the stream may pass through any number of nodes, and the same packet will be collected at multiple nodes, which not only wastes valuable sampling resources, At the same time, the sampling accuracy is reduced. The repeated packets are reduced as much as possible, and the problem is avoided by the superposition relationship between the nodes, which not only improves the sampling accuracy, but also improves the efficiency of the upper application of the collector).

The model contains a number of sub-constrained sub-problems. We consider the complexity and feasibility while quantifying, and it is actually difficult to directly solve the solution to the problem. Therefore, we decompose the problem model into three sub-models: node selection, Slot number allocation, and Slot sequence arrangement. For each sub-problem, we model it independently and use an independent algorithm to solve the optimal solution or approximate optimal solution. In the end, the approximation of the problem can be effectively obtained. This problem is not only a variant of multiple backpacks, but also adds enough constraints on the issue of multiple backpacks. Asking to solve this problem is an NP-hard problem. Therefore, we break it down into three parts to approximate the problem.

4. Optimal

In this section, we decompose the IMM into three sub-problems: K sampling point Selection, Slot time allocation, node cooperative sampling optimization, and corresponding three heuristic algorithms are proposed to form the approximate optimal solution of the IMM model.

4.1. K Sampling Point Selection

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. Therefore, we do not care about the effect of the number of slots in (1) on the comprehensive influence of the nodes (next section will show you how to make a reasonable allocation of slot number with selected points).The optimization model for this subproblem is equation (7), where $D_i =$.The $maxdsdada$ part of equation (7) expresses that any flow will only have a direct influence value for one of all nodes, and its direct influence on other nodes will be subtracted.In other words, any flow is privatized by a node during sampling nodes selection.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 selection);That is, for the direct influence value of the candidate R_i in the k round, the flow set included is $F_i^{cs} = F_i - \bigcup_{m=1}^{k-1} F_m^c$;If R_i is selected in this round, the flows in F_i^{cs} is R_i exclusive;In the $k+1$ round selection, even if the 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, so that the sum of the comprehensive influences of the K nodes is guaranteed to be maximized, so the K nodes are the optimal solutions.D_i^k indicates the direct influence of the candidate node R_i in the k-th round selection, which is calculated as equation (8). Equation (11) describes the iterative calculation formula for the comprehensive influence of the candidate nodes for each round of selection.Algorithm 1 gives the solution process, and Fig3 also shows the change process of direct influence in each round of elections.Finally, the K node set R^s and the Fcs set corresponding to each node can be obtained.

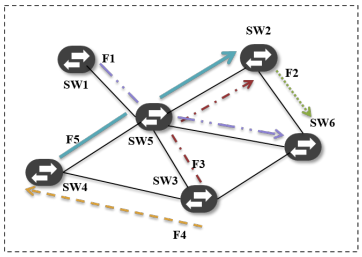
$$D_i^k = \left| F_i - \bigcup_c^{k-1} \widetilde{F_c} \right| / \left| F - \bigcup_c^{k-1} \widetilde{F_c} \right| \tag{7}$$

$$I_i^k = \alpha \cdot D_i^k + \beta \cdot S_i + \gamma \cdot H_i \tag{8}$$

Algorithm 1 Sampling Point Selection

Input:
The set of routers: R
The size of node will be selected: K
The current flow information matrix: M

define $R^s = \{\}$ // The Set of Selected Routers
for $k = 1$; $k < K$; $k++$ **do**
  for each $R_i \in R - R^s$ **do**
    if $I_i^k > max$ **then**
      $max = I_i^k$
      $SR = R_i$
    end if
  end for
  put SR to R^s
  mark SR as R_k in R^s
end for
return R^s



	S1	S2	S3	S4	S5	S6
F1	1	0	0	0	1	1
F2	0	1	0	0	0	1
F3	0	1	1	0	1	0
F4	0	0	1	1	0	0
F5	0	1	0	1	1	0
Betweenness Centrality	1	3	2	2	3	2

Figure 3: Intermediary center based on the number of streams

4.2. Allocation of Time Slot

After the node is selected, the number of Slots needs to be allocated to these nodes when the constraint (10)

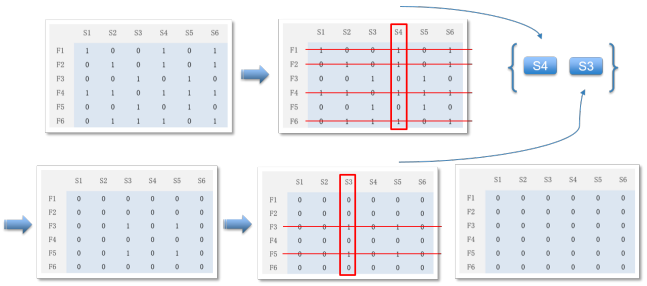


Figure 4: Sampling Point Selection

is satisfied and the Slot sequence between nodes is not considered.For each sample node, each value is assigned to a certain value, we still use the comprehensive influence to quantify the value of the node.The optimization model of the sub-problem is the formula (8), and the direct influence judgment of the formula is still the formula (3).The optimization model of the subproblem is the formula (8), which is still judged by the formula (3).And use the flow set privatized by the node to calculate the direct influence of the node in the unit time t D_i, $D_i = \frac{\sum_{f_k \in F_i^{cs}} P\{N_p^k(t)>0\}}{|F^c|}$.For the cost w_i of the node in unit time t, the dynamic influence D_i in equation (7) still uses F_i^c to calculate not F_i^{cs}. 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.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 (3) is satisfied, another growth trend will be presented,and he trend depends on S_i and H_i. We have described the reasons in the construction of the IMM model.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-impact nodes preferentially allocate Slot;

4.3. Order of Time Slot

Determining time series for each node, in Model(1), reflects the effects of overlapping relationships between nodes, such as the accuracy of the sampling of the system and the repetition rate of the sample packets. In the second chapter, we use the high-impact privatization of overlapping flows to approximate the effect of the precision caused by overlapping flows between nodes. Therefore, in this section, we reduce the repetition rate of the packet by optimizing the sampling sequence of each node Slot.

This optimization problem can be defined by the following formula(16). Where $\left|\widetilde{S_i} \bigcap \widetilde{S_j}\right|$ represents the number of overlaps of the time slots of the two nodes, and $\left|\widetilde{F_i} \bigcap \widetilde{F_j}\right|$ represents the number of overlapping flows between the two

Algorithm 2 Impact Priority Polling Allocation Slots

Define $CNT[1..K] = 0$; $\widetilde{I}[1..K] = 0$;
while $C > 0$ OR C is different from the last round **do**
  for *each* $R_i^s \in R^s$ **do**
    if R_i satisfy$\delta\,(v_i, |CNT[i]|)$ the $ELSE$ condition **then**
      $\widetilde{I}[i] = \frac{1}{T/t} \cdot (\beta \cdot S_i + \gamma \cdot H_i)$
    else
      $\widetilde{I}[i] = \alpha \cdot \frac{v_i \cdot 1}{|F^c|} + \frac{1}{T/t} \cdot (\beta \cdot S_i + \gamma \cdot H_i)$
    end if
  end for
  Descending sorting $\widetilde{I}$;
  for $i = 1$; $i < K$; $i++$ **do**
    if $C >= w_i$ **then**
      $CNT[i]++$
      $C = C - w_i$
    end if
  end for
end while
return CNT

nodes of ij. Therefore, this formula embodies the overlap area of the flow in the entire sampling system.

$$\min \sum \left(\left|\widetilde{S_i} \bigcap \widetilde{S_j}\right| \cdot \left|\widetilde{F_i} \bigcap \widetilde{F_j}\right|\right), \forall i, j \wedge i \neq j \qquad (9)$$

From the definition of the problem, the search backtracking method can be used to solve the optimal solution, 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 the steps. In the previous section, the CNT array has been calculated. At the beginning of the algorithm, the M^slot two-dimensional array is initialized to store the placement relationship between R_i and s^l: $M^slot[i][l] = 1$, which represents the Ri node sampling at s^l. In each round, a slot is selected for the node whose CNT is not 0, and the total number of coverages of the stream generated after the selected slot is selected is the minimum value of all the optional slots. Therefore, through each round, for each node whose CNT is not 0, a Slot that minimizes the total number of coverages of the current entire system stream is arbitrarily placed until all the Slots of the nodes are placed (all CNT are 0).), we get an approximate optimal solution. In the solution process, the $H[l]$ array is used to store the total number of coverages of each Slot stream, and when s^l is selected once, $H[l]$ is updated. After the node R_i selects s^l, the new flow cover total of s^l is calculated as: $H[l] = \sum_{j=1 \wedge j!=i}^{K} (\left|\widetilde{F_i} \bigcap \widetilde{F_j}\right| \cdot M^{slot}[j][l]) + H[l]$. That is, in equation (17), $\left|\widetilde{S_i} \bigcap \widetilde{S_j}\right| = 1$, and j is the node that has been selected for placement in s^l. The demonstration of the whole algorithm is shown in Fig. 6. In the example, the approximate solution we solved by this algorithm is 35, and the optimal solution is 33.

Algorithm 3 Order of Time Slot Based on Greedy

Input: M , S , c_j

while $CNT[i] > 0, \exists i \wedge i = 1, 2..., k$ **do**
  for $i = 1;\ i <= K;\ i++$ **do**
    if $CNT[i] > 0$ **then**
      $Min = Max\ Integer$
      for $l = 1;\ l < \frac{T}{t};\ l++$ **do**
        if $M^{slot}[i][l] = 0$ **then**
          $temp = \sum_{j=1 \wedge j!=i}^{K} (\left|\widetilde{F_i} \bigcap \widetilde{F_j}\right| \cdot M^{slot}[j][l]) + H[l]$
          if $temp < Min$ **then**
            $Min = temp;\ Sp = l$
          end if
        end if
      end for
      $M^{slot}[i][Sp] = 1;\ CNT[i] --;\ H[Sp] = Min$
    end if
  end for
end while
return M^{slot}

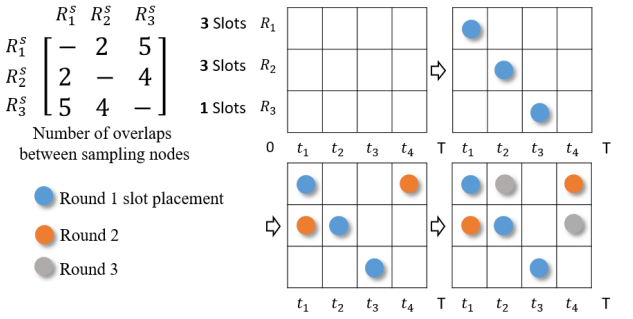
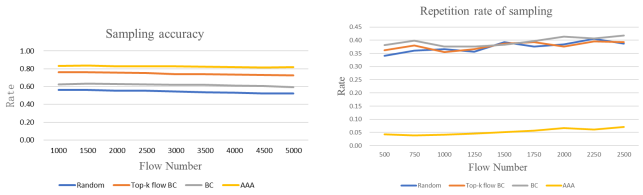


Figure 5: Illustrating sampling node slots placement based on greedy algorithm

5. Experiments and results

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



(a) comparison of sampling accuracy (b) comparison of repetition rate

Figure 6: comparison with respect to different algorithms

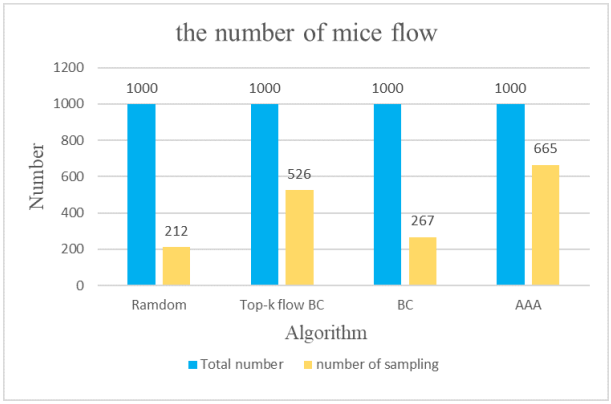


Figure 7: comparison of different algorithms in number of mice flow

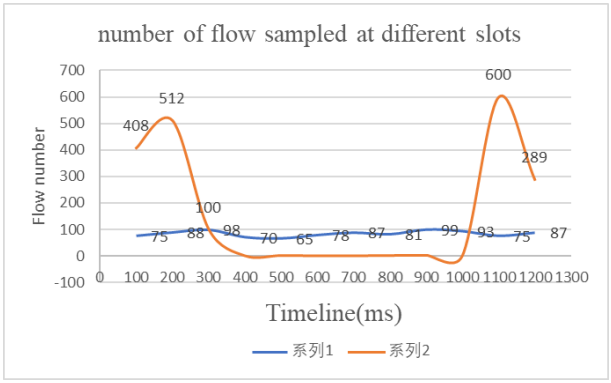


Figure 8: comparison of sampling flow number at different slot

6. Conclusion

- Lab environment

- 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. Christodoulopoulos, G. Landi et al, NEPHELE: An End-to-End Scalable and Dynamically Reconfgurable Optical Architecture for Application-Aware SDN Cloud Data Centers, in *IEEE Communitions 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.