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 increasing, 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.

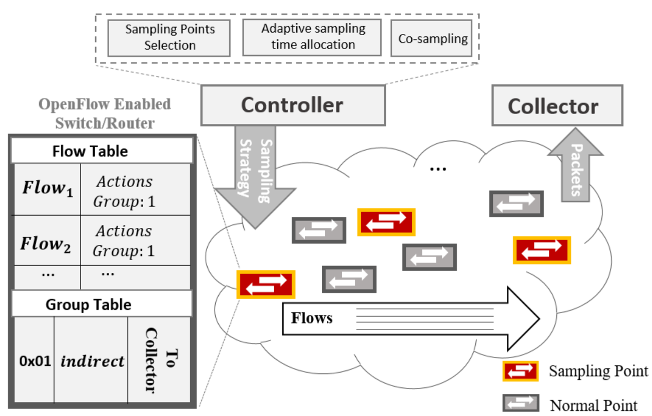


Figure 1: The System Architecture Of AAA

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.

In large-scale networks, maximizing the sampling accuracy of Flow-level is a huge challenge to satisfy the low intrusiveness of the network and the maximum limitation of packet Collector (IDS e.g) analysis capability.Therefore, a highly agile and adaptive algorithm is needed to establish a sampling strategy for the real-time situation of the current network. When the network size is large, assuming the number of switches is n, K (K ¡ = n) nodes are generally selected as sampling nodes, and K is determined according to the actual situation.If the bandwidth of the collector is C packets/T, the total number of packets sampled by all nodes in a cycle T does not exceed C. Therefore, given C and K, the reasons that affect the sampling accuracy include the selection of sampling nodes and the allocation of sampling time of each node. Another reason is the effective ratio of sampling.Packet repetition rate refers to different sampling nodes capture the same packet at almost the same time, called E, then the effective ratio is 1-E. Collecting the same packet will not only occupy limited sampling resources and limit sampling accuracy, but also reduce the efficiency of the upper application (IDS e.g).Therefore, each sampling node must coordinate their own sampling time to reduce the overlapping sampling time as much as possible, thereby reducing the global packet repetition rate and improving the global sampling accuracy.

3.1. IMM Model

In order to maximize the sampling accuracy of Flow-level, we build the model from three aspects: node selection, time allocation and cooperative strategy among nodes. We transform the maximum sampling accuracy problem to the maximum area coverage problem, as shown in Figure 2.In a sampling period T, the flow set F in the whole network is just like an area depicted by a red solid line, then the set F_i^C of the flow detected by node R1 is equivalent to the grey area, which is denoted as the direct value of the node.The red dotted area represents the potential value of the node, which is a new data flow that may be captured by the node in a T. These streams arrive only after the downward sampling strategy, so they are not perceived by the sampling algorithm.The overlap area of the grey area represents the same flow detected between the current nodes, as in $F_i^C \bigcap F_j^C$. The overlapping area of the red dotted line is the same new flow detected by each node.The sum of the direct value and the potential value represents the total value of nodes in a T. The larger the number of nodes detected, the greater the value of nodes. Therefore, the Flow-Level sampling problem can be directly converted to the area coverage maximization problem.If the sampling time of two nodes overlaps and $F_i^c \bigcap F_j^c ! =$, this is considered to be flow overlap, then the number of overlapping streams should be subtracted when calculating the overall coverage area, which reflects the cooperative nature of the sampling nodes.Therefore, in order to maximize the sampling accuracy of flow, the more flow overlap in the system, the smaller the sampling time between nodes overlap.However, the red area of nodes is unknown when sampling strategy is formulated. Even by analyzing the flow arrival distribution model, we can not know the number of overlapping flows in the future of each node. Therefore, the potential value of nodes needs a separate or personal quantization.

We propose a quantitative approach based on influence, which transforms the direct value of nodes into direct influence and the potential value into potential influence. We evaluate the potential value of nodes from two dimensions:the power in a network[2] and the number of former flows.They all represent the potential influence of the node, which represents the potential value of the node, that is, the greater the potential influence, the greater the value that the node may create in a unit time.We used standardized mediation centrality [2] to measure the power of nodes in the network, and recorded the network impact of $R_i$within a T as S_i.If the direct impact of $R_i$is greater than R_j, but $S_j$is higher than S_i, then even if the current flow deteced by $R_i$is bigger, the probability of the data flow passing through $R_j$is greater later.The percentages of previous flows reflects the activity of nodes throughout the network lifecycle, which we call historical influence. And we labeled the historical influence of $R_i$within a T as H_i, $H_i = TF_i/TF$.So the potential influence of $R_i$on periodic T is the combination of S_i and H_i.

Assuming that the arrival of packets of a flow $f_i(i=1,2...,k)$ obeys the Poisson distribution with λ_i. if any switch captures at least one packet from the same f_i, f_i is considered to have been successfully captured. The probability of f_i being captured in unit time t is $P\{N_p^k(t)>0\}$.If you assign t to the node R_i, the direct value of R_i is $D_i=\sum_{f_k\in F^c} P\left\{N_p^k\left(t\right)>0\right\}$.Therefore, the acquisition rate of $R_i$is $D_i/|F^c|$, which we call direct influence.The quantification of direct influence is based on the flow betweenness centrality. Overall, R_i has a combined effect of $\alpha\cdot D_i+\beta\cdot S_i+\gamma\cdot H_i$ over a t, and $\alpha+\beta+\gamma=1$.

We give the Maximum Impact Model (MIM) as (1). The goal of this 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, so as to maximize the influence of the system in the period T, and then maximize the flow-level sampling accuracy.Since time is continuous, let t is the unit time, $t<=T$. And let $L=T/t$ represents the number of time slots s^l(l=1,2...,L) in the T.$\widetilde{S_i}$ represents the set of slots allocated to R_i.The larger the size $|\widetilde{S_i}|$ of slot set, the greater the combined influence of nodes.The relationship between the combined influence of nodes and $|\widetilde{S_i}|$ is given in (2). When $D_i\cdot|\widetilde{S_i}|>|F_i^c|$, it means that nodes can capture all the flows currently detected.Then its direct impact will not continue to increase with the increase of $|\widetilde{S_i}|$, but the potential impact will continue to increase. In addition, when a flow is collected by different nodes in the same slot, the actual number of flow in the whole system does not increase.Therefore, when the direct force of all nodes is accumulated in the left part of equation (1), it may be calculated repeatedly.U is a set of routers which are assigned the s^l for detecing the f_k,as (3).When $|U|\geq 1$,the times $\psi(f_k,s^l)$ of f_k counted repeatedly at s^l is $|U|-1$(4).So the direct influence calculated repeatedly of whole network is $\alpha\cdot P\{N^k(t)>0\}\cdot\psi(f_k,s^l)/F^c$ via the f_k at the s^l ,in which α is a weight coefficient.In summary, the whole (1) represents the summation of the comprehensive influence of all nodes, and then subtracts the direct influence calculated repeatedly of all nodes to obtain the total influence of the system. Just like area coverage, the area of overlapping parts is calculated many times without increasing the actual area coverage.

Formula (5) (6) describes the constraints that the MIM model should satisfy. The number of sampling nodes $f(\widetilde{S_i})<=K$, and the number of sampling packets $w_i\cdot|\widetilde{S_i}|<=C$ sampled in period T. In formula (6), w_i is the price of R_i in unit time t. If the current rate of R_i is $w_i packets/T$, Formula (7) describes the calculation process of w_i, and $w_i*[(\beta\cdot S_i+\gamma\cdot H_i/\alpha\cdot D_i]$ represents an estimate of the possible arrival rate of the node. The cost of the new flow that the nodes may arrive in the T is not negligible. We use the ratio of influence to estimate the equal ratio.

We explained the quantization process of the IMM model and illustrated it as $Fig.2$.The goal of the MIM model is to allocate reasonable $|\widetilde{S_i}|$ sets for all $R_i$under constraint conditions.As $Fig2$ 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, 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 effective ratio of the system is improved.

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

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

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

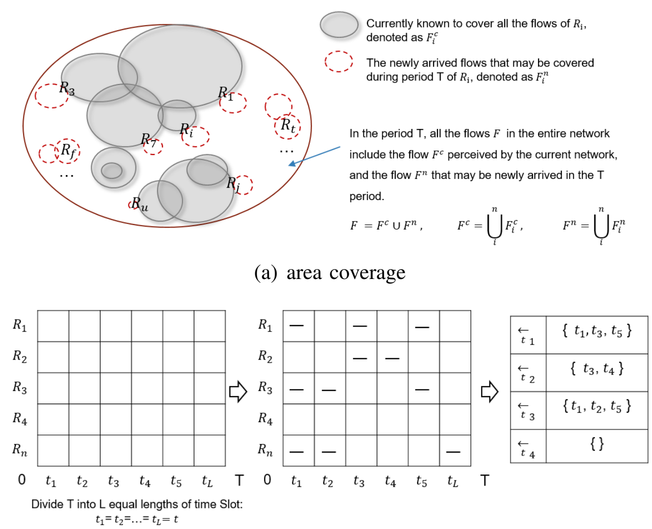
$$\psi\left(f_k,s^l\right)=\begin{cases}|U|-1, |U|\geq 1\\ 0 \quad\quad\;\, , |U|=0\end{cases} \tag{4}$$

subject to:

$$\sum\nolimits_i^n f(\widetilde{S_i})\leq K, f(\widetilde{S_i})=\begin{cases}1,\left|\widetilde{S_i}\right|\geq 1\\ 0,\left|\widetilde{S_i}\right|=0\end{cases} \tag{5}$$

$$\sum\nolimits_i^n w_i\cdot\left|\widetilde{S_i}\right|\leq C \tag{6}$$

$$w_i=\frac{t}{T}\cdot[\nu_i\cdot(1+\frac{\beta\cdot S_i+\gamma\cdot H_i}{\alpha\cdot D_i})] \tag{7}$$



(a) area coverage

(b) time slot allocation

Figure 2: overview of model

4. Optimal

In this section, we decompose the IMM into three subproblems: 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. The goal of the optimization model 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.The optimization model for this subproblem is (7), where $D_i = |F_i^c|/|F^c|$,and $\varphi(i)$ is 0 means that R_i is not selected, 1 means R_i is selected; $\widehat{U} = \{R_i, f_k \in F_i^c \wedge \varphi(i) = 1\}$ indicates how many selected nodes cover f_k.Similar to the IMM model, the former part sums the comprehensive influence of the selected nodes, and the latter part subtracts the influence part of the repeated calculations.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.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 $\widetilde{F}_i^{cs}$ to calculate the direct influence, where $\widetilde{F}_i^{cs} = F_i^c - \bigcup_{m=1}^{k-1} F_m^{cs}$,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 $\widetilde{F}_i^{cs}$ is R_i exclusive,then $\widetilde{F}_i^{cs}$ 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, 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 $D_i^k = \left|F_i - \bigcup_c^{k-1} \widetilde{F_c}\right| / \left|F - \bigcup_c^{k-1} \widetilde{F_c}\right|$. 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 F^{cs} set corresponding to each node can be obtained.

$$max[\sum_{i=1}^{n}(\alpha \cdot D_i + \beta \cdot S_i + \gamma \cdot H_i) \cdot \varphi(i) - \alpha \cdot \sum_{f_k \in F^c} \widehat{\psi}(f_k, i)] \quad (8)$$

$$subject: \sum_{i=1}^{n} \varphi(i) = K \quad (9)$$

$$\widehat{\psi}\left(f_k, s^l\right) = \begin{cases} \left|\widehat{U}\right| - 1, & \left|\widehat{U}\right| \geq 1 \\ 0 &, \left|\widehat{U}\right| = 0 \end{cases} \quad (10)$$

$$I_i^k = \alpha \cdot D_i^k + \beta \cdot S_i + \gamma \cdot H_i \quad (11)$$

Algorithm 1 Sampling Point Selection

```
define R^s = {} // The Set of Selected Routers
for k = 1; k < K; k++ do
    for each R_i ∈ R − R^s do
        if I_i^k > max then
            max = I_i^k
            R_k^s = R_i
        end if
    end for
    put R_k^s into R^s
    F_K^cs = F_i^c − ⋃_{m=1}^{k−1} F_m^cs
end for
return R^s
```

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) 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-influence nodes preferentially allocate Slot.After each round of allocation, if a node satisfies the ELSE condition of formula (3), the comprehensive influence of the node in unit time t is $\beta \cdot S_i + \cdot H_i$,which does not satisfy the formula (3) For ELSE condition, 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.

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$\delta\left(v_i, |CNT[i]|\right)$ 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

4.3. Order of Time Slot

In the IMM model, we describe the effect of cooperative sampling between nodes on sampling accuracy and the effective ratio of sampling. As shown in Fig. (3), the cooperation between nodes is reflected in the order of sampling Slots of each node.For this sub-problem, formula (9) 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 $|S_i \bigcap S_j|$ indicates the number of times they are sampled under the same Slot;$|F_i^c \bigcap F_j^c|$ indicates the number of flows that thet cover the same.Therefore, the number of flows that are resampled between two nodes can be expressed as: $|S_i \bigcap S_j| \cdot |F_i^c \bigcap F_j^c|$.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.

$$\min \sum \left(\left|\widetilde{S_i} \bigcap \widetilde{S_j}\right| \cdot \left|\widetilde{F_i} \bigcap \widetilde{F_j}\right|\right), \forall i, j \wedge i \neq j \tag{12}$$

This problem can be solved by using the search backtracking 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: $M^s lot[i][l] = 1$, which represents R_i node is sampled at s^l.In each round, an idle slot is selected for all nodes of $i, CNT[i]! = 0$, 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 is selected, an approximate optimal solution is obtained.Fig.6 demonstrates the process. In the example, the approximate solution we solved by this algorithm is 6 and the optimal solution is 5.

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**
          $temp = \sum_{j=1 \wedge j!=i}^{K}\left(\left|\widetilde{F_i} \bigcap \widetilde{F_j}\right| \cdot M^{slot}[j][l]\right) + H[l]$
          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}

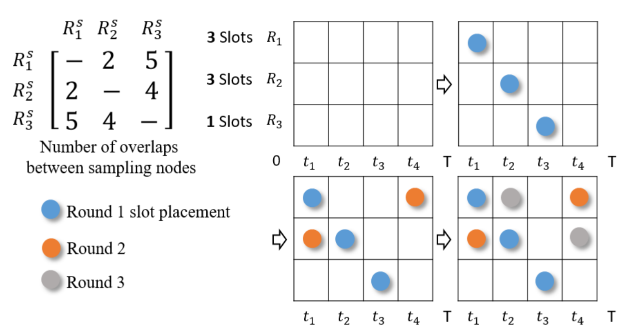
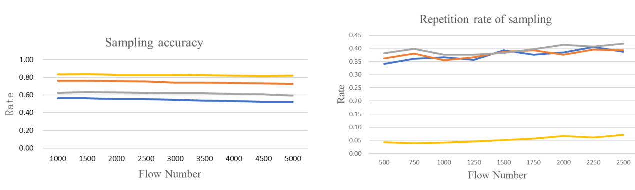


Figure 3: Illustrating sampling node slots placement based on greedy algorithm

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 4: comparison with respect to different algorithms

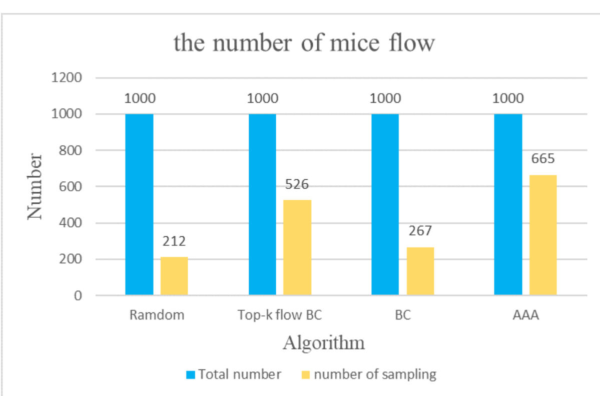


Figure 5: comparison of different algorithms in number of mice flow

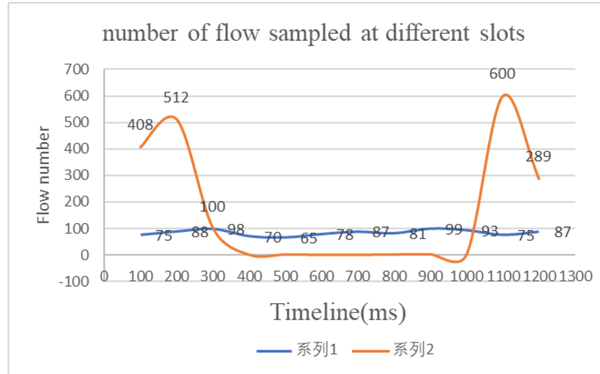


Figure 6: comparison of sampling flow number at different slot

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.