PRAM: Priority-aware Flow Migration Scheme in NFV Networks

Zili Meng\textsuperscript{1,2}, Jun Bi\textsuperscript{1,2,3}, Chen Sun\textsuperscript{1,2,3}, Anmin Xu\textsuperscript{1,2,3}, Hongxin Hu\textsuperscript{4}
\textsuperscript{1}Institute for Network Sciences and Cyberspace, Tsinghua University
\textsuperscript{2}Tsinghua National Laboratory for Information Science and Technology (TNList)
\textsuperscript{3}Department of Computer Science, Tsinghua University
\textsuperscript{4}Clemson University
mengzl15@mails.tsinghua.edu.cn, junbi@tsinghua.edu.cn, \{c-sun14, xam14\}@mails.tsinghua.edu.cn, hongxih@clemson.edu

ABSTRACT
We propose PRAM, a priority-aware flow migrating scheme to address the problem of which flows to migrate during dynamic scale in or out of network functions in NFV networks. We introduce the concept of Priority-Weighted Migration Time (PWMT), and use a greedy algorithm to minimize the PWMT, in order to exert the minimum impact on latency-sensitive flows. Simulation results show that PRAM can reduce the PWMT by 5.61× to 18.48× compared with the random strategy.

CCS Concepts
\begin{itemize}
  \item Networks \rightarrow Network performance modeling; Network control algorithms; Network management;
\end{itemize}

Keywords
Network Function Virtualization; flow scheduling; network management algorithm

1. INTRODUCTION
The combination of Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) offers the potential to enhance service flexibility and reduce overall costs. With the ability to dynamically redistribute packets among multiple instances of a Virtualized Network Function (VNF), NFV + SDN enables SDN controllers to migrate some flows on a certain VNF instance away when scale-in or scale-out is needed.

Some recent research efforts have been devoted to designing mechanisms for safe and efficient flow state migration among VNF instances [1, 2, 4]. However, deciding which flows to migrate is also a significant problem in VNF scaling. For example, the additional latency caused by migration (tens of milliseconds in TFM [4]) may be unacceptable for some flows that demand ultra-low latency within NFV networks (e.g. algorithmic stock trading flows), while acceptable for flows without such strict latency demand (e.g. P2P downloading flows). Thus, under the same circumstances, migrating the P2P flows rather than stock trading flows could ensure that fewer SLAs for tenants are breached.

To address the above problem, in this poster, we propose PRAM, a PRiority-Aware flow Migration scheme, which targets on both selecting enough flows to migrate to alleviate the VNF load, and exerting little impact on latency sensitive flows. To quantify the factor of flow latency sensitivity, we enable network operators to set priorities for flows. Higher priority indicates a higher flow latency sensitivity. The specific values of priorities can be also automatically set based on a network management algorithm concerning the SLAs with tenants, the classification of flows, and administrator’s traffic strategy. We introduce the concept of Priority-Weighted Migration Time (PWMT) that considers both flow latency sensitivity and flow size in our design. To fulfill the target of PRAM, we design a greedy algorithm to minimize the PWMT of the flows to migrate. Our evaluation demonstrates that PRAM can reduce the PWMT by 5.61× to 18.48× compared with the random strategy.

2. DESIGN
As mentioned above, we assign high priorities for flows with high latency sensitivity and introduce PWMT for each flow, which is the product of migration time and flow priority. We intend to select enough flows to alleviate the VNF load, and migrate flows with lower latency sensitivity, i.e. priority, during migration. Thus, we minimize the sum of PWMT, which measures the impact of migration on these flows, as our objective.
Suppose there are $N$ flows in total, we assign flow $i$ with priority $p_i$, size $s_i$, and migration time $t_i$. Based on the current load of the VNF and operator policies, suppose the total flow size to migrate is $S_{mig}$. We denote $x_i$ as an indicator of whether flow $i$ is migrated. $x_i = 1$ means flow $i$ will be migrated, while $x_i = 0$ means flow $i$ will not be migrated. We express the optimization as:

$$
\min \sum PWMT = \sum x_i p_i t_i \quad s.t. \quad \sum x_i s_i > S_{mig}
$$

According to [1, 4], we derive two major observations:

**Observation 1:** The migration time for each flow increases linearly with the number of flows to migrate in total. Results in [4] show a high linearity with a correlation coefficient of 0.991.

**Observation 2:** The migration time for each flow is irrelevant to the flow size. The flow migration latency is incurred by moving flow states. The migration time is only related to VNF internal state storage pattern and irrelevant to the flow size.

Based on the observations above, we rewrite the migration time $t_i = \cdots = t_N = A(\Sigma x_i) + B$, where $\Sigma x_i$ is the number of flows to migrate, and $A$ and $B$ are two constants only related to the VNF. The modeling is meaningful as $A$ and $B$ are in the same order of magnitude [4]. Thus the objective function could be transformed into:

$$
\min (A \sum x_i + B) \sum x_i p_i \quad s.t. \quad \sum x_i s_i > S_{mig}
$$

This optimization problem could be described as a 0-1 Integer Programming (0-1 IP) problem, which is NP-hard. Due to the possibly frequent flow migration, it is unacceptable to use IP to enable dynamic flow migration. To minimize the objective function, we tend to move flows with (1) lower priority, which exerts a lower impact on highly sensitive flows, and (2) larger size, which requires fewer flows to migrate and thus reduces the total migration time for all flows. However, the two factors could conflict in some cases, e.g. some large flows could be highly sensitive to the latency. Therefore, we define the parameter of flow size per priority (SP2), i.e. $s_i / p_i$, to express the two factors, and use *Greedy Algorithm* to quickly find a satisfying solution. We split the algorithm into the following three steps:

**Step 1:** We calculate the SP2 of each flow in the VNF.

**Step 2:** As our greedy strategy, we pick the flow with the highest SP2 to satisfy the two factors above.

**Step 3:** PRAM checks whether enough flows have been selected to migrate (Size of flows that have been selected should be larger than $S_{mig}$). If not, the algorithm goes back to Step 2. The cycle continues until enough flows have been selected for migration.

### 3. EVALUATION

In evaluation, we use the flow size distribution in the datacenter [3] for simulation. The priorities of flows are set randomly from 1 to 100 with uniform distribution. We take the results of migration time in [4] and use MATLAB to do simulations of 1,000 flows for 20 times.

**4. CONCLUSION AND FUTURE WORK**

In this poster, we have proposed a priority-aware flow migration scheme, PRAM, which considers the latency sensitivity when selecting flows to migrate for VNF scaling. We introduce the PWMT to quantify the latency sensitivity, and design a greedy algorithm to minimize the sum of PWMT of all flows to migrate as our optimization objective. According to our evaluation, PRAM outperforms the random flow selection strategy significantly.

As our future work, we will consider more evaluation metrics and do more precise experiments to evaluate the effectiveness of PRAM. We will also enhance the robustness of PRAM and consider more complex states.

### 5. REFERENCES


