Enhancing DRL-Based Multi-Path Routing with Subflow Identification in 6G Packet-over-Optical Networks

H. Shakespear-Miles[1], N. Koneva[2], S. Barzegar[3], M. Ruiz[1], A. Sánchez-Macian[2], and L. Velasco[1*]

[1] *Optical Communications Group (GCO), Universitat Politecnica de Catalunya (UPC), Barcelona, Spain*
[2] *Dept. Ing. Telematica, Universidad Carlos III de Madrid, Spain*
[3] *Barcelona Super Computing Center (BSC), Barcelona, Spain*
[*] *luis.velasco@upc.edu*

Abstract: Packet-over-optical transport networks in the 6G era must handle massive traffic volumes and dynamic traffic patterns while ensuring near-real-time quality of service (QoS). To address this challenge, we extend our previous deep reinforcement learning (DRL)-based multi-path flow routing to include subflow identification. Specifically, we leverage the Count-Min Sketchs (CMS) to identify and estimate heavy subflows (HsF). This ensures packets within the same subflow follow a single path to prevent packet reordering and additional delay. Our results demonstrate the use of CMS for subflow identification and impact of the solution on end-to-end delay, highlighting the importance of precise subflow identification in maintaining network performance.

1. Introduction

Real-time monitoring and management of large-traffic volumes is essential in high-speed metro networks to maintain reliability, ensure quality of service (QoS). To this end, machine learning (ML) methods have proven valuable for enhancing resource efficiency, adding adaptability to optical networks, and supporting near-real-time autonomous network management [1]. In our earlier work [2], we introduced a deep reinforcement learning (DRL)-based approach for decentralized and autonomous routing of packet flows in packet-over-optical networks, focusing on meeting strict QoS requirements—particularly minimizing end-to-end (e2e) delay. The DRL agent dynamically determined how traffic should be distributed across multiple available paths. These paths were provisioned by a centralized software-defined networking (SDN) controller, ensuring sufficient capacity for the flow while offloading real-time routing decisions to the DRL system. This setup enabled near-real-time routing adjustments while reducing the SDN controller's operational load.

To prevent packet reordering, an issue that can increase delay [3], each traffic flow was a collection of subflows, with each subflow consistently routed along a single path. However, our earlier design did not include a method for identifying individual subflows within a flow. This limitation can be addressed using the Count-Min Sketch (CMS), a memory-efficient probabilistic data structure capable of identifying the most significant subflows (heavy subflows, or HsFs) by estimating their sizes and assigning identifiers. Prior work [4] has shown that CMS can be effectively implemented within P4-programmable switches. Additionally, P4 switches can deal with complex routing tables, where subflows can be easily labelled and forwarded to the desired path, as well as implement in-band network telemetry (INT) to collect detailed end-to-end delay metrics [5].

In this paper, we build upon our previous contributions [1][6] by providing additional details and results for autonomous subflow routing architecture that integrates CMS for subflow detection and labelling, and the subflow partitioning module, which uses the DRL-derived routing policy to optimally assign subflows to paths, ensuring adherence to e2e delay constraints.

2. Architecture and Traffic Flow Scenario

The system design and use case are illustrated in Figure 1. A substantial volume of network traffic, composed of smaller subflows, begins at site A and is routed toward site B using three designated paths (p1, p2, and p3). These routes are established and maintained by an SDN controller, which ensures enough bandwidth for the overall traffic demand is provided. At the ingress point (site A), a P4-configured switch applies routing policies to distribute the subflows across the available paths, embedding in-packet metadata using INT. When reaching site B, the P4 switch captures this metadata and processes it to compute network performance indicators such as e2e delay, updated at fine-grained intervals (for instance, every few hundred milliseconds). At coarser time intervals—i.e., once per second—a telemetry module aggregates these metrics and forwards them to a source-side agent.

Within site A, a subflow identification module operates directly within the P4 switch, continuously analyzing incoming packets to detect dominant subflows and estimate their respective bitrates. It is important to note that a single traffic flow may consist of both long-lived subflows and numerous transient ones with low bitrate or short duration, which may not be distinctly identifiable. To maintain flow consistency, all packets belonging to the same flow must follow the same route. Thus, any portion of the traffic flow that remains indistinguishable is treated as a single, indivisible subflow.

979-8-3315-9777-1/25/$31.00 ©2025 IEEE

Additionally, site A hosts a DRL based routing engine atop the P4 switch. This engine periodically determines the optimal routing strategy—represented as a distribution percentage of the total traffic across the three available paths—based on real-time flow metrics and delay statistics. This strategy is then refined by a subflow partitioning component, which allocates specific subflows to paths in a way that approximates the DRL-defined traffic distribution, relying on insights from traffic measurements and subflow identification.

However, the actual distribution of subflows may diverge from the ideal routing policy, especially when the flow includes heavy subflows with large volumes. These discrepancies can lead to imbalanced traffic distribution and potential QoS degradation due to increased delays. To address this, continuous monitoring of flow composition and analysis of mismatches between actual subflow distribution and the intended routing policy is crucial. A management module evaluates these deviations and provides feedback to the SDN controller, enabling it to dynamically adjust path configurations to enhance flow handling and QoS compliance.

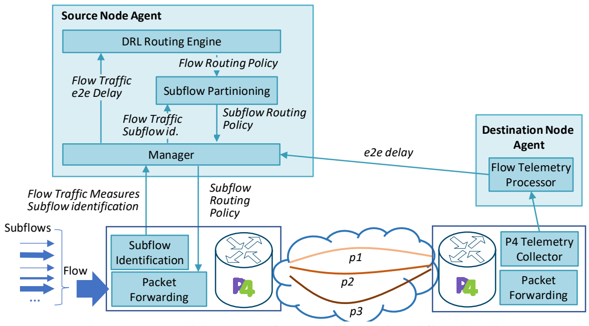

Figure 1: Architecture for subflow identification in multipath routing scenario

Algorithm 1: Subflow partitioning algorithm for multiple subflows

Input: X(t), *P*, *H*	**Output**: *sFR*
1:	$X(t+1) \leftarrow$ predict($X(t)$)
2:	**for each** $id \in H$ **do** $H[id] \leftarrow H[id] \times (X(t+1)/X(t))$>
3:	$nH \leftarrow X(t+1)$ - sum(H)
4:	$H[0] \leftarrow nH$
5:	$H \leftarrow$ sort(H, DESC)
6:	$sFR \leftarrow \{\}$
7:	**for each** $id \in H$ **do**
8:	$path\_id \leftarrow$ find_min_diff(sFR, P, H)
9:	$sFR \leftarrow sFR \cup \{$<$id$, $H[id]$, $path\_id$>$\}$
10:	**return** sFR

3. Methodology

The proposed solution contains two main parts described in this section: i) identification of subflows using CMS and ii) the partitioning of subflows based on DRL decision making.

CMS is a space-efficient, probabilistic algorithm for estimating element frequencies in high-volume data streams, using significantly less memory than traditional counting methods. It comprises a $d \times W$ matrix, where d is the number of hash functions and W the number of columns. For each incoming packet, d independent hash functions map the flow ID to a column in each row, incrementing the corresponding counters. That is, the cell $[d, h_d]$ in the matrix is increased by one per hash to reflect the packet's contribution to the flow.

To estimate a flow's frequency, its ID is hashed with the same d functions, retrieving the corresponding counters from each row. The minimum of these values is returned as the estimate. While efficient, the method introduces estimation errors due to hash collisions inherent in the sketch's structure.

This error can also be estimated as CMS is classified as a (ε, δ)-approximation algorithm, where ε is maximum relative error, δ is the probability of exceeding the maximum error. The estimation error is bounded by $\varepsilon_{|\mathrm{Count}|}$ with a probability of $1 - \delta$. The relationships between these parameters are expressed as:

$$d = \lceil \ln(1/\delta) \rceil \quad (1)$$

$$W = \left\lceil \frac{e}{\varepsilon} \right\rceil \quad (2)$$

where e represents Euler's number. Additionally, the upper bound on the estimation error can be expressed as Equation 3 where $\widehat{C_k}$ is the estimation from the CMS and C_k is the true count.

$$\widehat{C_k} - C_k \leq \varepsilon \cdot k \quad (3)$$

Once the HsFs are identified by the CMS algorithm, the partitioning algorithm can proceed to decide which available path they should be assigned to. The full partitioning algorithm is defined in Algorithm 1. It receives the currently observed aggregated flow traffic, *X(t)*, as well as inputs both from the CMS identification and the DRL decision making. Specifically, the flow routing policy, *P*, decided by the DRL decision making that includes the IDs of the available paths and the percentage of traffic that should be routed through each path and the estimated size of each HsF identified, *H*. Since the routing policy will be implemented in the next time interval, the partitioning algorithm uses the aggregated traffic and a polynomial predictor to make a prediction on the total traffic in the next time interval (line 1 of Algorithm 1). Then, the estimated size of each HsF is used to determine what proportion of the aggregated traffic it comprises, and then are scaled by the previous polynomial prediction to obtain the estimated HsF size in the next time interval (line 2). To determine the non-HsF traffic volume, the cumulative proportional HsF traffic is subtracted from the aggregated traffic in the subsequent time interval (line 3) and then stored (line 4). All the subflows (HsF and aggregated non HsF) are sorted in descending order so that the subflows with larger traffic requirements are allocated first (line 5). Each subflow is considered as a candidate

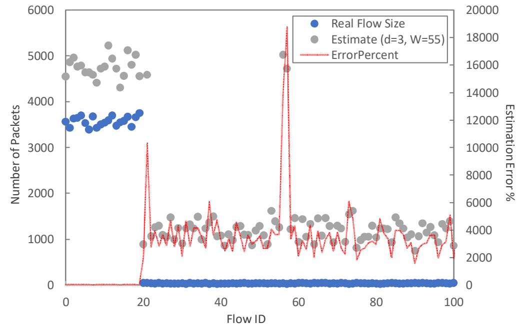

Figure 2: CMS estimation error for subflows

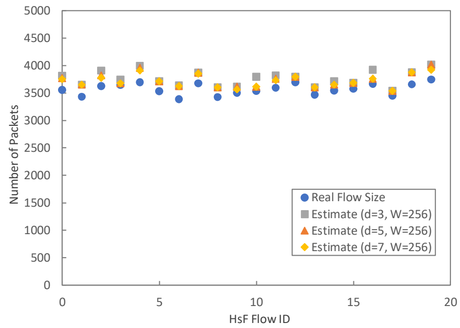

Figure 3: CMS estimation error for 20 HsFs and W=256

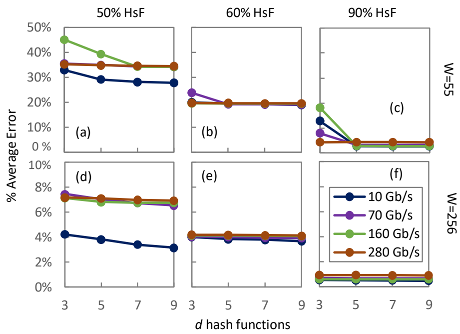

Figure 4: CMS identification error for varying d, W, and proportions of HsFs.

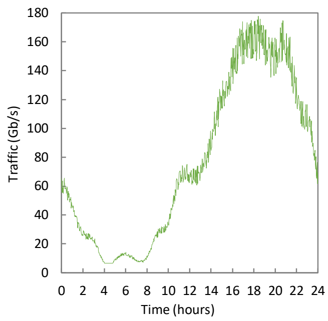

Figure 5: Daily traffic profile

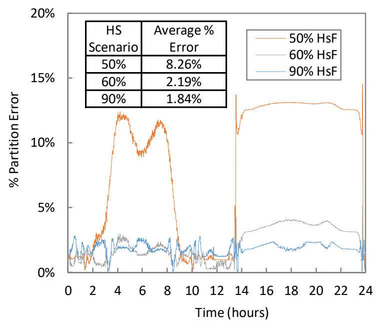

Figure 6: Partitioning error over 3 path routing

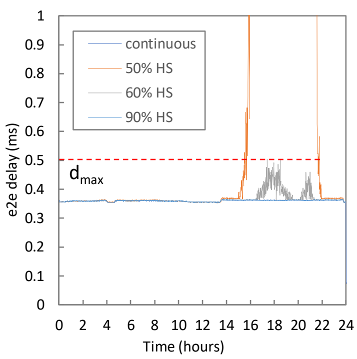

Figure 7: e2e response for HsF partioning

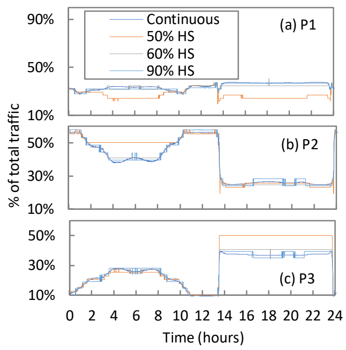

Figure 8: % of traffic sent through each path

Table 1: Memory consumption in bits for varying *d* and *W*, CMS dimensioning

Memory (bits)		*W*	
		55	256
d	3	2640	12288
	5	4400	20480
	7	6160	28672
	9	7920	36864

for assignment to each available path, with evaluation of the capacity difference that would occur upon the subflow being assigned.

The path is then selected that would result in the smallest capacity difference without exceeding the routing policy decision and taking into account the subflows already assigned to that path (line 8). Once a path is selected it is stored with the other assigned subflows (line 9). Once each subflow has been assigned, the resulting subflow routing policy is returned for implementation (line 10).

4. Illustrative Results

A Python-based simulator was used to evaluate the proposed methods and algorithms. The simulator consists of a traffic generator that creates packet traces for multiple subflows, each following a daily traffic pattern. Subflow volumes are randomly determined based on simulation parameters. Packet sizes follow the trimodal distribution [64, 596, 1500] with probabilities [0.333, 0.167, 0.5], as described in [4], and inter-arrival times are adjusted to match the desired time series.

Traffic scenarios use average rates from [4], testing four rates—10, 70, 160, and 280 Gb/s—combined with three HsF traffic proportions (50%, 60%, and 90%), totalling 12 scenarios. Each scenario includes 20 HsFs, allowing analysis of both total traffic impact and HsF contribution to partitioning performance. For each scenario, the CMS

uses a 0.1-second monitoring window, initialized at $t_0 = 0$ with all counters set to zero. Packets are counted until $t_1 = 0.1$ s, after which different values of depth d and width W are tested to evaluate CMS accuracy.
To illustrate, consider the scenario with 10 Gb/s average rate and 50% HsF traffic. Figure 2 shows actual flow sizes, CMS estimates, and estimation errors using d =3, W=55 for the first 100 subflows. The 20 HsFs appear on the left side of the plot, exhibiting significantly lower error than non-HsF flows. This highlights CMS's effectiveness in identifying heavy hitters—a key factor in Algorithm 1. Figure 3 shows CMS error across varying d values (with fixed W) for the same scenario. While some HsFs benefit from higher d, estimation accuracy remains largely consistent across different depths.

Next, Algorithm 1 can be evaluated with the CMS estimations for each of the traffic scenarios with varying d and W values in Figure 4. As expected, the larger the proportion of HsF the lower the error. Additionally, a W of 256 produces a lower error than W=55 for all HsF proportions. Next the operation of the CMS with partitioning algorithm can be studied for the period of a day where the daily traffic profile is shown in Figure 5. For all experiments a polynomial predictor of degree 2 and window size of 10 was used for traffic predictions. The resulting partitioning error is plotted in Figure 6. where 50% HsF proportion shows much higher partitioning error overall and especially when the traffic is at near its minimum(4-8h) and maximum (16-22h). This is confirmed by the average error values shown in the inset table, where 50% HsF proportion produced an average of 8.26% partitioning error. The effect of the partitioning error on the e2e delay can be studied in Figure 7, where both 60% and 90% HsF proportions are able to maintain the delay below the d_{max} requirement, and 50% HsF violates this requirement when the total traffic is large. To further understand the partitioning error and delay response the percent of traffic for each proportion of HsFs can be plotted and compared to the continuous or ideal partition provided by the DRL engine in Figure 8. The 60% and 90% HsF proportions allow the traffic percentages to closely approximate the continuous, but the 50% HsF proportion forces 50% of the traffic to be routed through the same path. For example, an instruction of 40% traffic from the DRL engine would be forced to 50% through a single path and a corresponding reduction in traffic to the other paths translating to increased e2e delays as there may not be enough capacity in that path for 50% of the traffic.

It should be noted that the total memory consumption of the CMS structure is calculated by d x W x16 bit counters. The memory consumption for all configurations considered is shown in Table 1. The trade-off between memory consumption and accuracy should be considered, as the accuracy of the estimation improves greatly when increasing W from 55 to 256.

5. Conclusions

In conclusion, the critical role of accurate HsF identification and partitioning was shown as an enabler to effective autonomous multi-path flow routing for maintaining e2e delay requirements was shown. By identifying the HsFs and ensuring they are sent through the same path packet reordering issues are avoided at the destination. The CMS was shown to provide high accuracy identification across different scenarios, emphasizing the importance of the dimensioning for both accuracy and memory usage. The portioning algorithm was shown the be effective for approximating the DRL engine instructions in the cases where the proportion of HsFs was greater than 50%. However, estimates from the HsF identification can then be used by the SDN controller to provide paths with capacity meeting the flow traffic requirements to mitigate this effect.

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme SEASON (G.A. 101096120) and from the ICREA Institution.

References

[1] D. Rafique and L. Velasco, "Machine Learning for Optical Network Automation: Overview, Architecture and Applications," JOCN, 2018.
[2] S. Barzegar *et al.*, "Autonomous Flow Routing for Near Real Time Quality of Service Assurance," TNSM 2024.
[3] J. Lin *et al.*, "Packet Reordering in the Era of 6G: Techniques, Challenges, and Applications," Electronics, 2023.
[4] J.A. Hernandez *et al*. "Count Min sketches for Telemetry analysis of performance in P4 implementations," ONDM 2024.
[5] P. Gonzalez *et al.*, "Distributed Multi-Agent System fed with Telemetry Data for Near-Real-Time Service Operation," OFC, 2024.
[6] H. Shakespear-Miles *et al.*, "Near-Real-Time Autonomous Multi-Path Flow Routing with Subflow Identification", OFC, 2025.