APPENDIX

A. Proof of Theorem V.1

Proof. We begin by proving the no over-estimation error property of Algorithm 1, followed by a corresponding proof for Algorithm 2.

Algorithm 1 (Proof by Case Analysis).

For Algorithm 1, we prove by case analysis that

$$\forall e_i \in \mathcal{S}, \forall j \in M, \hat{P}_{e_i}^{j} \leq P_{e_i}^{j}$$

Base Case ($j = 0$): Before seeing any flows($j = 0$), $\hat{P}_{e_i}^{0} = 0$ and $P_{e_i}^{0} = 0$. Thus $\hat{P}_{e_i}^{j} \leq P_{e_i}^{j}$.

Inductive Hypothesis: Assume that for the (j-1)-th time window W_{j-1}, $\hat{P}_{e_i}^{j-1} \leq P_{e_i}^{j-1}$ holds for every flow e_i. We now show it also holds at the time window W_j.

Case 1: The arriving flow $x \neq e_i$ at time window W_j. The persistence estimate for e_i either stays the same or decreases. Therefore, $\hat{P}_{e_i}^{j} \leq \hat{P}_{e_i}^{j-1} \leq P_{e_i}^{j-1} \leq P_{e_i}^{j}$, and $P_{e_i}^{j-1} \leq P_{e_i}^{j}$ because the true persistence cannot decrease over time. Hence, $\hat{P}_{e_i}^{j} \leq P_{e_i}^{j}$.

Case 2: The arriving flow $x = e_i$ at time window W_j and e_i is already stored in the bucket. Then the true persistence $P_{e_i}^{j} = P_{e_i}^{j-1} + 1$ (if $W_j \neq W_{e_i}$) or $P_{e_i}^{j} = P_{e_i}^{j-1}$ (if $W_j = W_{e_i}$) , thus $\hat{P}_{e_i}^{j} \leq \hat{P}_{e_i}^{j-1} \leq P_{e_i}^{j-1} \leq P_{e_i}^{j}$.

Case 3: The arriving flow $x = e_i$ at time window W_j but e_i is not currently stored in the bucket–and there is at least one empty cell. The Algorithm decides to store e_i in an arbitrary empty cell, initializing $\hat{P}_{e_i}^{j} = 1$. Meanwhile, the true count for e_i has increased by 1 at this step, meaning $P_{e_i}^{j} = P_{e_i}^{j-1} + 1$. Hence we have $\hat{P}_{e_i}^{j} \leq 1 \leq P_{e_i}^{j}$.

Case 4: The arriving flow $x = e_i$ at time window W_j but e_i is not currently stored in the bucket–and there is no empty cell. In the Algorithm, e_i tries to replace the flow in the bucket which has the minimum persistence $\hat{P}(e_{min})$. The replacement happens when $\hat{P}_{e_{min}} - 1 = 0$ with probability $\frac{1}{\hat{P}_{e_{min}}+1}$.

- If replacement is successful, then $\hat{P}(e_i) = 1$. Meanwhile P_{e_i} is at least 1, so $\hat{P}_{e_i}^{j} \leq P_{e_i}$.
- If replacement is unsuccessful,then $\hat{P}_{e_i}^{j} = 0$. Meanwhile $P_{e_i} \geq 1$, so still $\hat{P}_{e_i}^{j} \leq P_{e_i}$.

In all cases, we see $\hat{P}_{e_i}^{j} \leq P_{e_i}^{j}$. By the principle of step-by-step induction, this property holds at every time window W_j and for every flow e_i.

Algorithm 2 (Proof by Contradiction).

We prove for Algorithm 2, by contradiction, that

$$\forall e_i \in \mathcal{S}, \forall j \in M, \hat{P}_{e_i}^{j} \leq P_{e_i}^{j}$$

Assumption: $\exists e_i$, s.t. $\hat{P}_{e_i} > P_{e_i}$.

By the definition of $\hat{P}_{e_i}$ in Algorithm 2, $\hat{P}_{e_i}$ is computed via persistence increments that occur only when e_i arrives, and it may be reset to 0 if it is replaced by another flow. A replacement occurs when $\hat{P}_{e_i} - 1 = 0$ with a probability $1 - e^{-\frac{\alpha \times inactivity}{\hat{P}_{e_i}}}$. Let I_{e_i} be the total number of times the algorithm increments the persistence for e_i over the entire time horizon. Then, by definition,

$$\hat{P}_{e_i} \leq I_{e_i}$$

Moreover, each increment to e_i occurs only under one of the following conditions: (1) e_i is stored and in this time window $W_j \neq W_{e_i}$, (2) e_i is not stored, but there is empty cell in the bucket, or (3) e_i successfully replace an flow in the bucket according to the Algorithm 2. Therefore, the total number of persistence increments I_{e_i} cannot exceed the true persistence of e_i. Therefore,

$$I_{e_i} \leq P_{e_i}$$

Combining the two inequalities, we obtain

$$\hat{P}_{e_i} \leq I_{e_i} \leq P_{e_i}$$

This directly contradicts our assumption that $\hat{P}_{e_i} > P_{e_i}$. Hence, our contradiction shows $\forall e$,

$$\hat{P}_{e_i} \leq P_{e_i}$$

This completes the proof.

Therefore, we have established the no over-estimation error property for both Algorithm 1 and Algorithm 2. ☐

B. Definition of (ϵ, δ)-persistence

Definition A.1 ((ϵ, δ)-persistence)**.** *Given a small positive number ϵ and an error rate threshold $\delta > 0$, the probability $\Pr\left(|P_{e_i} - \hat{P}_{e_i}| \geq \epsilon M\right)$ represents the likelihood that the error in the estimated persistence, $P_{e_i} - \hat{P}_{e_i}$, exceeds ϵM. An algorithm satisfies (ϵ, δ)-persistence if, $\forall e_i \in \mathcal{S}$, the following condition holds:*

$$\Pr\left(|P_{e_i} - \hat{P}_{e_i}| \geq \epsilon M\right) \leq \delta$$

C. Proof of Theorem V.2

Proof. Let e be a flow, and let $N(G)$ denote the number of its arrivals during a time window W_j of duration G. Since arrivals follow a Poisson process with rate λ, the probability of observing k arrivals is:

$$\Pr(N(G) = k) = \frac{(\lambda G)^k e^{-\lambda G}}{k!}, \quad k \geq 0.$$

Consider a persistent flow e_i. When e_i arrives and is mapped to a bucket $B(v, q)$ already occupied by a different flow $e' \neq e_i$, the persistent value is either unchanged, decreased by 1, or reset to 1.

Let W_j denote the (random) time window in which e_i first enters a bucket. Define:

- Δ_1: number of windows before W_j in which e_i appeared but was not placed into any bucket.
- Δ_2: number of windows at or after W_j in which the persistence count of e_i is decreased due to hash collisions—specifically, when another flow is hashed into the same bucket and a replacement attempt occurs. The probability of such a replacement attempt is defined by the reciprocal of the current counter value of e_i plus one, even if the replacement ultimately fails.

We assume that once a persistent flow e_i enters a bucket, it is not evicted from the bucket by other flows. Also, by Theorem V.1, no overestimation occurs. Thus, the estimated persistence satisfies:

$$\hat{P}_{e_i} = P_{e_i} - \Delta_1 - \Delta_2$$

Where P_{e_i} is the true persistence of flow e_i.

Applying Markov's inequality for any $\epsilon > 0$:

$$\Pr\left(|P_{e_i} - \hat{P}_{e_i}| \geq \epsilon M\right) = \Pr\left(\Delta_1 + \Delta_2 \geq \epsilon M\right) \quad (1)$$

$$\leq \frac{\mathbb{E}[\Delta_1 + \Delta_2]}{\epsilon M} \quad (2)$$

We now bound $\mathbb{E}[\Delta_1]$. Let P_a be the probability that e_i hashes into a bucket with another flow but fails to replace it. For each flow e, the probability that in each of the r arrays, in one time window, there are hash collisions in the bucket to which flow e is mapped:

$$\Pr\left(\bigcap_{e' \in \mathcal{S} \setminus \{e\}} \{h_v(e) = h_v(e')\}\right) \leq \left[1 - \left(1 - \frac{1}{w}\right)^{n-1}\right]^r \triangleq \Omega^r \quad (3)$$

Hence,

$$P_a \leq \frac{1}{w} \cdot \Omega^r \quad (4)$$

Let $X_{0,j}(e_i)$ be the number of windows before W_j in which e_i appeared at least once. Since e_i follows a Poisson process with rate λ, we have:

$$\mathbb{E}[X_{0,j}(e_i)] = j(1 - \Pr\left(N(G) = 0\right)) = j(1 - e^{-\lambda G})$$

$$\mathbb{E}[\Delta_1] = P_a \cdot \mathbb{E}[X_{0,j}(e_i)] \leq \frac{j}{w}\Omega^r(1 - e^{-\lambda G}) \quad (5)$$

Now, we bound $\mathbb{E}[\Delta_2]$. After flow e_i enters the bucket, based on the algorithm, in each such window, at most one successful collision-induced replacement can decrease the persistence count of e_i by 1. And the bucket of e_i is chosen at random $\frac{1}{r}$. Let P_b be the probability that a distinct flow e' hashes into the same bucket as e_i, and make the count reduce by 1:

$$P_b \leq \frac{1}{rw} \cdot \frac{\Omega}{\hat{P}_{e_i} + 1} \leq \frac{\Omega}{2rw} \quad (6)$$

Define the indicator variable: $I_k = 1$, which means at least one other flow is hashed to the same bucket as e_i in the window W_k. Then,

$$\Delta_2 = \sum_{k=j}^{M-1} I_k, \quad \text{and} \quad \mathbb{E}[\Delta_2] = \sum_{k=j}^{M-1} \mathbb{E}[I_k].$$

Each other flow has an expected number of arrivals λG per window. There are $n-1$ other flows, so the total expected number of arrivals is $(n-1)\lambda G$. The number of arrivals hashed to the same bucket as e_i is:

$$\text{Poisson}((n-1)\lambda G P_b).$$

Thus, the probability that at least one such arrival occurs is:

$$\mathbb{P}(I_k = 1) = 1 - e^{-(n-1)\lambda G P_b}.$$

Based on the algorithm, when a collision happens, e_i must not have appeared in the same window. The probability is $e^{-\lambda G}$. Then, for each window W_k from j to $M-1$, the expectation that at least one such collision occurs is:

$$\mathbb{E}[\Delta_2] \leq (M-j)e^{-\lambda G}\left(1 - \exp\left(-\frac{(n-1)\lambda G \Omega}{2rw}\right)\right) \quad (7)$$

Combining both parts:

$$\begin{aligned}
&\Pr\left(|P_{e_i} - \hat{P}_{e_i}| \leq \epsilon M\right) \leq \frac{\mathbb{E}\left(\Delta_1 + \Delta_2\right)}{\epsilon M} \\
&\leq \frac{j}{w\epsilon M}\Omega^r(1 - e^{-\lambda G}) \\
&+ \frac{(M-j)e^{-\lambda G}}{\epsilon M}\left(1 - \exp\left(-\frac{(n-1)\lambda G \Omega}{2rw}\right)\right) \\
&\underset{(a)}{\leq} \frac{1}{w\epsilon} + \frac{1}{\epsilon \lambda G}\left(1 - \exp\left(-\frac{(n-1)\lambda G}{2rw}\right)\right) \\
&= \frac{1}{\epsilon w \lambda G}\left(\lambda G + w\left(1 - \exp\left(-\frac{(n-1)\lambda G}{2rw}\right)\right)\right) \quad (8)
\end{aligned}$$

Where (a) uses the fact that $e^{-x} \leq \frac{1}{1+x} \leq \frac{1}{x}$ for $x > 0$, bounds $\Omega^r \leq 1$ and $j \leq M$.

Hence, we obtain:

$$\delta = \frac{\lambda G + w\left(1 - \exp\left(\Phi\right)\right)}{\epsilon w \lambda G}$$

where

$$\Phi = -\frac{(n-1)\lambda G}{2rw}$$

□

D. Proof of Theorem V.3

Proof. We adopt the same notation as in the proof of Theorem V.2. The difference in the analysis begins from the estimation of $\mathbb{E}[\Delta_2]$, which reflects the number of windows where the persistence count of a flow e_i is decremented due to collision-induced replacement.

After flow e_i is successfully inserted into a bucket, the optimized algorithm updates its persistence count only if no new packet from e_i arrives in a window and a collision occurs with another flow e'. The probability that such a flow e' hashes to the same bucket and causes a decrement in the persistence count of e_i is denoted by P_b.

By the design of the optimized algorithm, we have:

$$\begin{aligned}
P_b &\leq \frac{\Omega}{rw} \cdot \left(1 - \exp\left(-\frac{\alpha \cdot \mathit{inactivity}}{\hat{P}_{e_i}}\right)\right) \\
&\leq \frac{\Omega \alpha \cdot \mathit{inactivity}}{rw\hat{P}_{e_i}} \quad (9)
\end{aligned}$$

where $\mathit{inactivity}$ denotes the number of consecutive windows immediately preceding the current window during which e_i did not appear. Let W_j be the current window, and suppose e_i last appeared in window W_k. Then, $\mathit{inactivity} = j - k$.

The number of windows until e_i reappears follows a geometric distribution with success probability

$$p_1 = 1 - e^{-\lambda G},$$

where λ is the arrival rate of e_i and G is the duration of a time window. Thus, the expected inactivity is:

$$\mathbb{E}[j-k] = \frac{1}{p_1} = \frac{1}{1-e^{-\lambda G}}.$$

Substituting this into the bound for P_b from (9), we obtain:

$$P_b \leq \frac{\Omega\alpha}{rw(1-e^{-\lambda G})}. \tag{10}$$

Each other flow $e' \neq e_i$ has an expected λG arrivals per window, and there are $n-1$ such flows. Thus, the expected number of arrivals that may cause a decrement in a given window is Poisson-distributed with rate $(n-1)\lambda G P_b$. The probability that at least one such arrival occurs, while e_i does not appear in the same window, is:

$$\begin{aligned}\mathbb{E}[\Delta_2] &\leq (M-j)e^{-\lambda G}\left(1-\exp\left(-(n-1)\lambda G P_b\right)\right) \\ &\leq (M-j)e^{-\lambda G}\left(1-\exp\left(-\frac{\alpha(n-1)\lambda G\Omega}{rw(1-e^{-\lambda G})}\right)\right),\end{aligned} \tag{11}$$

where M is the total number of windows.

To bound the total deviation in the estimated persistence count, we use the same framework as before:

$$\begin{aligned}&\Pr\left(\left|P_{e_i} - \hat{P}_{e_i}\right| > \epsilon M\right) \leq \frac{\mathbb{E}[\Delta_1+\Delta_2]}{\epsilon M} \\ &\leq \frac{1}{\epsilon w \lambda G}\left(\lambda G + w\left(1-\exp\left(-\frac{\alpha(n-1)\lambda G}{rw(1-e^{-\lambda G})}\right)\right)\right) \\ &\underset{(a)}{\leq} \frac{1}{\epsilon w \lambda G}\left(\lambda G + w\left(1-\exp\left(-\frac{\alpha(n-1)}{rw}\right)\right)\right)\end{aligned} \tag{12}$$

(a) uses the fact that $\frac{1}{1-e^{-\lambda G}} \geq \frac{1}{\lambda G}$.

Hence, we conclude that the error probability under the optimized algorithm is bounded by:

$$\delta_{\text{opt}} = \frac{\lambda G + w\left(1-\exp(\Phi_{\text{opt}})\right)}{\epsilon w \lambda G}$$

where

$$\Phi_{\text{opt}} = -\frac{\alpha(n-1)}{rw}$$

$\square$