Supplementary Materials - StarStream: Live Video Analytics over Space Networking

Anonymous Authors

1 CORRECTIONS

In our original manuscript, there is a typo in the objective of optimization Problem (1). Since we define α and β as positive weighting parameters, the sign before βQ_k should be "-" instead of "+". We apologize for any confusion this may have caused. The correct one should be written as follows.

$$
\begin{aligned}
&\underset{c_{k1},\cdots,c_{k2}}{\arg\max} \quad \sum\nolimits_{k=k1}^{k2} \alpha A_k(c_k) - \beta Q_k \\
&s.t. \quad \begin{cases}
\bar{b}_k = \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} b_t \, dt \\
t_k = t_{k-1} + \sum_j e_j(c_k) + \dfrac{\sum_j d_j(c_k)}{\bar{b}_k} + \Delta t_k \\
Q_k = Q_{k-1} + (t_k - t_{k-1}) - L_k \\
c_k \in C, \qquad \forall k = k1, \cdots, k2
\end{cases}
\end{aligned}
\tag{1}
$$

2 MEASUREMENT CONFIGURATION

Section 3 of the original manuscript reports the measurement results of the (frame rate, resolution) combination that yields the highest accuracy for each target bitrate and video. The specific (frame rate, resolution) combinations are shown in Table 1.

Table 1: The (frame rate, resolution) combination that yields the highest accuracy for each target bitrate and video, where 1280 **represents the resolution** 1280×720 **and** 1920 **represents the resolution** 1920×1080**.**

Target bitrate	hw1	hw2	street	beach
1.5 Mbps	(15, 1280)	(15, 1280)	(5, 1280)	(5, 1920)
3 Mbps	(15, 1280)	(15, 1920)	(3, 1920)	(15, 1920)
4.5 Mbps	(15, 1280)	(15, 1920)	(5, 1920)	(15, 1920)
6 Mbps	(15, 1280)	(15, 1920)	(5, 1920)	(15, 1920)
7.5 Mbps	(15, 1280)	(15, 1920)	(15, 1920)	(15, 1920)
9 Mbps	(15, 1280)	(15, 1920)	(15, 1920)	(15, 1920)

3 DYNAMIC PROGRAMMING ALGORITHM

In this section, we give a detailed description of the dynamic programming algorithm used by the *shift-guided configuration optimizer* of StarStream.

Let $\mathcal{L}$ denote the set of candidate GOP lengths, and C denote the set of candidate configurations. For each GOP length $l \in \mathcal{L}$ and configuration $c \in C$, the profiled frame size vector is $\vec{d}(c,l) = \{d_1(c,l),\ d_2(c,l), \cdots\}$, where $d_j(c,l)$ denote the profiled frame size of the j^{th} frame in the GOP. Then, we define the set of all profiled frame size vectors as $\mathcal{D} = \{\vec{d}(c,l) \mid \forall c \in C \text{ and } \forall l \in \mathcal{L}\}$. Similarly, we define the set of all profiled frame encoding delay vectors as $\mathcal{E} = \{\vec{e}(c,l) \mid \forall c \in C \text{ and } \forall l \in \mathcal{L}\}$, where $\vec{e}(c,l) = \{e_1(c,l),\ e_2(c,l), \cdots\}$ represents the profiled frame encoding delay vector for configuration c and GOP length l. We further define the profiled accuracy set as $\mathcal{A} = \{A(c,l) \mid \forall c \in C \text{ and } \forall l \in \mathcal{L}\}$.

Assume that the *shift-guided configuration optimizer* is choosing configurations for the next 3 GOPs $\{GOP_{k+1}, GOP_{k+2}, GOP_{k+3}\}$. Let $\Gamma = \{\gamma_l \mid \forall l \in \mathcal{L}\}$ denote the set of current configuration accuracy scale factors. We further assume that according to the *throughput and shift predictor*, the corresponding chosen GOP length list is $L = [L_{k+1},\ L_{k+2},\ L_{k+3}]$, and the predicted upload throughput list is $B = [\bar{b}_{k+1},\ \bar{b}_{k+2},\ \bar{b}_{k+3}]$. Let t_k denote the global timestamp when the client finishes transmitting the last frame of GOP k, and Q_k is the corresponding camera buffer queue length. Then, Algorithm (1) details the dynamic programming algorithm used by the *shift-guided configuration optimizer* to solve Problem (1) over the next three GOPs. This algorithm maintains a set $\mathcal{S}(i)$ that contains all Pareto optimal configuration combinations from GOP_{k+1} to GOP_{k+i}. Each item in $\mathcal{S}(i)$ is a tuple of (t_{k+i}, Q_{k+i}, cumulative QoE, selected configuration list), representing a possible state when the client finishes transmitting the last frame of GOP_{k+i}.

The $sim(\cdot)$ function in line 8 of Algorithm (1) takes the current system state, the predicted throughput, the frame size estimates, and the encoding delay estimates as input, simulates the behavior of sending GOP_{k+i} over the network, and returns the timestamp when the transmission of the GOP is completed. The pruning method used by line 13 of Algorithm (1) is based on the following rule: if there exist $(t, Q, qoe, list) \in \mathcal{S}(i)$ and $(t', Q', qoe', list') \in \mathcal{S}(i)$ such that $t \leq t'$ and $qoe \geq qoe'$, then $(t', Q', qoe', list')$ will be removed from $\mathcal{S}(i)$. It should be noted that although the algorithm optimizes configurations over the next three GOPs, only the configuration of the next GOP (GOP_{k+1}) is applied according to the model predictive control (MPC) paradigm.

Algorithm 1: Dynamic Programming Algorithm

Input: $\mathcal{D}$; $\mathcal{E}$; $\mathcal{A}$; L; B; Γ; t_k; Q_k
Output: the chosen configuration for GOP_{k+1}

1 Initialize $\mathcal{S}(0) \leftarrow \{(t_k,\ Q_k,\ 0.0,\ [\,])\}$;
2 **for** $i = 1$ *to* 3 **do**
3 $\quad \mathcal{S}(i) \leftarrow \emptyset$;
4 $\quad$ Get current GOP length $L \leftarrow L[i-1]$;
5 $\quad$ Get predicted GOP upload throughput $\bar{b} \leftarrow B[i-1]$;
6 $\quad$ **foreach** *item* $(t, Q, qoe, c\_list) \in \mathcal{S}(i-1)$ **do**
7 $\quad\quad$ **foreach** *configuration* $c \in C$ **do**
8 $\quad\quad\quad t_{k+i} \leftarrow sim(t, c,\ \vec{d}(c,L),\ \vec{e}(c,L),\ \bar{b})$;
9 $\quad\quad\quad Q_{k+i} \leftarrow \max(0,\ Q + (t_{k+i} - t) - L)$;
10 $\quad\quad\quad qoe_i \leftarrow qoe + \alpha \cdot \gamma_L \cdot A(c,L) - \beta \cdot Q_{k+i}$;
11 $\quad\quad\quad list_i \leftarrow c\_list + [c]$;
12 $\quad\quad\quad \mathcal{S}(i) \leftarrow \mathcal{S}(i) \cup \{(t_{k+i},\ Q_{k+i},\ qoe_i,\ list_i)\}$;
13 $\quad$ Prune suboptimal items from $\mathcal{S}(i)$;
14 $best\_config\_list \leftarrow$ the item with the highest qoe in $\mathcal{S}(3)$;
15 return $best\_config\_list[0]$;