Perceptual Quality Maximization for Video Calls with Packet Losses by Optimizing FEC, Frame Rate and Quantization

Eymen Kurdoglu, Yong Liu and Yao Wang

Abstract

We consider video calls affected by bursty packet losses, where FEC is applied on a per-frame basis due to tight delay constraints. In this scenario, both the encoding (eFR) and the decoded (dFR) frame rates are crucial; a high eFR at low bitrates leads to larger quantization stepsizes (QS), smaller frames, hence suboptimal FEC, while a low eFR at high bitrates diminishes the perceptual quality. Coincidently, damaged frames and others predicted from them are typically discarded at receiver, reducing dFR. To mitigate frame losses, hierarchical-P coding (hPP) can be used, but at the cost of lower coding efficiency than IPP..I coding (IPP), which itself is prone to abrupt freezing in case of loss. In this paper, we study the received video call quality maximization for both hPP and IPP by jointly optimizing eFR, QS and the FEC redundancy rates, under the sending bitrate constraint. Employing Q-STAR, a perceptual quality model that depends on QS and average dFR, along with R-STAR, a bitrate model that depends on eFR and QS, we cast the problem as a combinatorial optimization problem, and employ exhaustive search and hill climbing methods to solve explicitly for the eFR and the video bitrate. We also use a greedy FEC packet distribution algorithm to determine the FEC redundancy rate for each frame. We then show that, for iid losses, (i) the FEC bitrate ratio is an affine function of the packet loss rate, (ii) the bitrate range where low eFR is preferred gets wider for higher packet loss rates, (iii) layers are protected more evenly at higher bitrates, and (iv) IPP, while achieving higher Q-STAR scores, is prone to abrupt freezing that is not considered by the Q-STAR model. For bursty losses, we show that (i) layer redundancies are much higher, rising with the mean burst length and reaching up to 80%, (ii) hPP achieves higher Q-STAR scores than IPP in case of longer bursts, and (iii) the mean and the variance of decoded frame distances are significantly smaller with hPP.

I. Introduction

As the average network speeds and the efficiency of various video codecs have steadily increased over the past decades, real-time video delivery applications, such as video calling and video conferencing, have become an integral part of our daily lives. One of the fundamental challenges in video transmission in real time, aside from simultaneously achieving high-rate and low-delay video transmission, is to provide resiliency against packet losses that degrade the users' quality-of-experience. Traditionally, packet loss resiliency is provided through a combination of layered video coding, forward error correction (FEC), and automatic repeat request (ARQ) at the sender side, along with error concealment techniques at the receiver side. ARQ introduces an additional round-trip delay in case of lost packets, and is therefore unsuitable due to the stringent delay requirement of real-time video delivery. This requirement also restricts the video data, i.e., frame, group of pictures (GoP), or intra-period, on which block-code based FEC can be applied, since the decoding cannot be completed without receiving sufficient number of source blocks. For video calls, applying FEC on frames removes the additional FEC decoding delays, since each frame can be encoded and decoded individually. However, the efficiency of block FEC codes is reduced when the number of source blocks is small. This is typically the case if the encoder does not reduce the frame rate at low target bitrates, resulting in frames that are not sufficiently large.

As indicated, adjusting the encoding frame rate (eFR) based on the sending bitrate (SBR) enables efficient frame-level FEC. More broadly, choosing the temporal resolution (eFR), amplitude resolution (quantization stepsize), and the spatial resolution (picture size), denoted jointly by STAR in [1], under an SBR constraint, is another challenge closely coupled with FEC. Different STAR combinations, each of which achieves the same video bitrate, leads to different video qualities for different video contents. In [2], it was shown that significant quality gains can be achieved by picking the STAR that maximizes the perceptual quality at a given SBR. In the presence of packet losses, however, FEC provides loss resiliency at the cost of additional bits, which limits the spatial and amplitude resolutions that can be achieved.

For error concealment, a simple technique, called frame-copying [3], is prevalent in industry applications [4]. This technique freezes the last decoded frame on screen in case of a damaged frame due to missing packets, until another frame can be decoded without errors. This is because missing packets typically affect a large region on the frame, and any error propagates to the following frames over extended regions due to the use of motion-compensated temporal prediction. Due to frame-copying error concealment, the decoded frame rate (dFR) is bounded by eFR, ultimately impacting the received video quality. Frame-copying may also cause severe video freezes if the underlying coding structure is IPP...I (IPP). The use of layered encoding, specifically temporal layering achieved through hierarchical-P coding (hPP) [5], mitigates the effect of packet losses and frame freezes, with minimal additional complexity. However, layered coding presents additional coding overhead compared to non-layered coding achieved through the more typical IPP...I coding (IPP).

Consequently, it is crucial to study the delicate interplay between the FEC, STAR, and different encoding methods for real-time video delivery. In this paper, we study the perceptual video quality maximization for video calls that are subject to varying bandwidths and packet losses, by jointly optimizing FEC, eFR and the quantization stepsize (QS), using either hPP or IPP coding.

We consider a constant picture size to reduce the problem complexity. We assume that the congestion control module periodically predicts the end-to-end available bandwidth in the network, and that the video frame delays are minimized as long as the SBR does not exceed the available bandwidth [6][7][8]. Similar to the previous studies, the sender allocates a fraction of the SBR for encoding the video, while the remaining bitrate is used for FEC that protects the compressed video stream through redundancy. Block codes are applied on each individual frame, with potentially different code rates. At the receiver side, we assume that the frame-copying error concealment is used. We utilize the Q-STAR model in [1] to evaluate the decoded video quality, which represents the perceptual quality as a function of STAR. When certain encoded frames are not decodable due to packet loss that cannot be recovered by FEC decoding, we simply approximate the decodable frame rate by the number of decodable frames per second. This is a reasonable assumption under the hPP structure and our unequal error protection strategy, to be detailed in Sections III, IV and V. We also leverage the R-STAR model proposed in [9], which relates the video rate with the encoding STAR, in our formulation. We cast the problem of optimizing eFR, the video bitrate, and the FEC redundancy rate for each frame as a combinatorial optimization problem, and solve it through a combination of exhaustive search and hill-climbing methods. We also propose a greedy algorithm that solves for the suboptimal redundancy rate for each frame under a given FEC bitrate constraint. Through simulations, we show that, for independent and identically distributed (iid) packet losses with up to 20% loss, the optimal FEC bitrate ratio can be approximated with an affine function of the packet loss rate, and the quality drop compared with the lossless case is minimal. Through simulations, we show that hPP provides significantly more regular frame intervals than IPP at small SBRs. For bursty packet losses, FEC redundancy ratios increase dramatically, even when the mean burst length is relatively small. In this case, hPP is able to deliver higher Q-STAR scores than IPP in a large SBR range, along with smaller mean and variance of decoded frame distances.

II. Related Work

Video transmission over unreliable networks requires both source and FEC coding, in general. Since infinite-length source or FEC coding blocks are not realizable, the *separation principle* [10], which states that the source and channel coding can be designed independently, cannot be used without performance loss in real-time video delivery applications. As a remedy, joint source-channel coding (JSCC), aiming to minimize the end-to-end source distortion, has been studied extensively for video delivery over lossy links or networks. Much of the existing work on this problem differs in (i) the video coding methods considered, (ii) the video distortion or quality metrics used, (iii) the total amount of video distortion caused by the unrecoverable packet losses, and (iv) the video data unit over which the chosen metric is estimated, and on which the FEC is applied. Traditionally, when layered video is considered, each layer is assumed to reduce the total MSE distortion by a certain amount that is known [11], or estimated [12], [13]. The additional distortion due to packet losses are mostly considered to be additive [12], however the nonlinear effect of bursty losses have been considered in [14]. Furthermore, the existing works mostly do not address the delay requirement of video calls, as the FEC is traditionally applied on a fraction of a single or multiple consecutive GoPs in [15], [16], where the receiver must wait for the entire GoP or sub-GoP to recover a frame, leading to either unacceptable delays or distortions. A few papers [17], [18] explored frame-level FEC strategies by maximizing the expected number of decoded frames under a total FEC bitrate constraint and assuming IPP coding structure. To the best of our knowledge, no prior work has studied frame-level FEC for hPP, nor studied the gains achieved by adapting the frame rate and FEC together.

III. Maximizing the Received Perceptual Quality

We consider a video call scenario between a source $\mathcal{S}$ and a destination $\mathcal{D}$ over an unreliable network, in which packets may get lost. The directed paths from $\mathcal{S}$ to $\mathcal{D}$ and $\mathcal{D}$ to $\mathcal{S}$ are called the forward and the backward paths, respectively. We assume that the available bandwidth on the forward path is predicted by a congestion control module at the sender, and based on this prediction, a safe maximum sending rate is determined to ensure minimal queuing delay[1]. We further assume that the packet loss events on the forward path are correlated in general. Similar to the previous work, we use the Gilbert model to describe the packet loss process on the end-to-end path. The arrival state of each packet, that is, whether it is lost or not, is represented by 0 or 1 in a discrete-time binary Markov chain, respectively. The corresponding state transition probabilities are

$$\Pr(0|1) = \xi_{0|1} \qquad \Pr(1|0) = \xi_{1|0}.$$

Then, the mean packet loss rate ϵ and the mean burst length λ are

$$\epsilon = \frac{\xi_{0|1}}{\xi_{0|1} + \xi_{1|0}} \qquad \lambda = \frac{1}{\xi_{1|0}}.$$

In this study, we consider only temporal layering to keep the layered encoding complexity and overhead at a minimum. Our goal is to explore the performance of both temporal-layered and non-temporal-layered encoding for video calls over lossy networks. We assume that the encoder inserts an I-frame every T seconds, which is called an *intra-period*. We employ the IPP structure in Fig. 1 for non-layered encoding, and the hPP structure in Fig. 2 for layered encoding with $L \geq 1$ temporal layers

[1] The design of such a sending bitrate control mechanism (congestion control) that ensures minimal frame delays is beyond the scope of this paper.

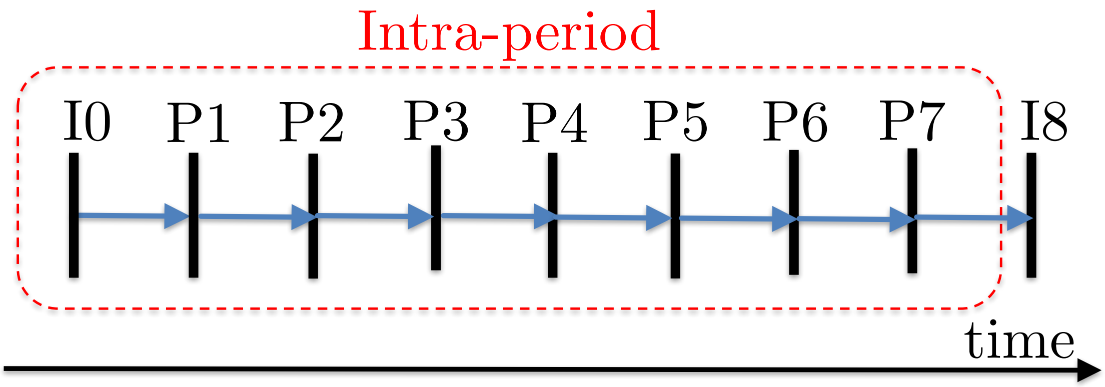

Fig. 1. IPP prediction with $N = 8$. Blue arrows indicate the prediction directions.

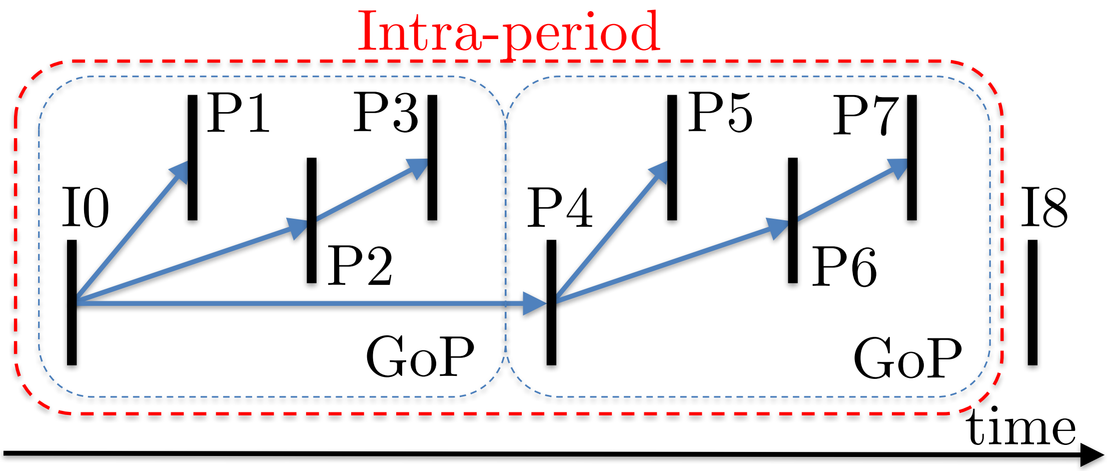

Fig. 2. hPP prediction with $N = 8$. Blue arrows indicate the prediction directions. In this example, $G = 4$, TL(1)={I0, P4}; TL(2)={P2, P6}; TL(3)={P1, P3, P5, P7}.

(TLs), where $G = 2^{L-1}$ is the length of a group of pictures (GoP). Note that, hPP reduces to IPP for $L = 1$. The number N of encoded frames that belong to a given intra-period is equal to $T \cdot f_e$, where f_e denotes the eFR in that intra-period.

Next, frame-level FEC is applied to protect the compressed video stream. The sender packetizes frame i of a given intra-period into $k_i = \lceil z_i/B \rceil$ network packets, where z_i is the size of the frame i and B is the maximum payload size. Then, Reed-Solomon coding [19] is applied across all packets of each individual frame i, by creating m_i redundancy packets based on k_i source packets. The redundancy rate for frame i is then $r_i = m_i/(k_i + m_i)$. By applying FEC across the packets of a single frame, instead of across packets belonging to multiple frames, we avoid further FEC decoding delay at the receiver.

When FEC fails and the current frame cannot be recovered, it is discarded along with its descendants in the dependency tree. In the meantime, the last video-decoded frame is frozen on the screen. This error concealment technique results in irregular distances between displayed frames that are free of artifacts, as opposed to regular intervals between displayed frames with noticeable artifacts that grow towards the end of the intra-period. Such frame-distance irregularity is less severe in hPP, since a playback freeze due to a frame loss in a temporal enhancement layer only lasts until the next lower-layer frame is decoded, providing higher resilience. For example, in Figure 2, if I0 and P1 are decoded and P2 suffers an unrecoverable packet loss, P2 and P3 are both discarded, regardless of whether P3 arrives or not. If P4 arrives, we can decode it from I0. In this case, P1, which has been kept on screen until now, will be replaced by P4. However, the same frame loss will make all subsequent frames P2-P7 non-decodable in Figure 1. Note that this error resilience is obtained at non-negligible expense of coding efficiency.

To measure the quality of the video displayed at the receiver, we use the perceptual quality model in [1]. According to this model, the perceptual quality of a decoded video sequence can be written as a function of the spatial, temporal and the amplitude resolution of the video. Then, keeping the spatial resolution constant, we have

$$
\begin{aligned}
Q(q, f_d) &= \mathrm{NQQ}(q) \times \mathrm{NQT}(f_d) \\
&= \frac{1 - e^{-\alpha_q \frac{q_{\min}}{q}}}{1 - e^{-\alpha_q}} \frac{1 - e^{-\alpha_f \left(\frac{f_d}{f_{\max}}\right)^{0.63}}}{1 - e^{-\alpha_f}}
\end{aligned}
\tag{1}
$$

where q is the QS and f_d is the dFR, whereas α_q and α_f are parameters that depend on the video characteristics, and $q_{\min}$ and $f_{\max}$ are minimum QS and the maximum frame rate values considered, for which $Q(q_{\min}, f_{\max}) = 1$.

We can now summarize the operation of the proposed video call system. Since the video bitrate control is usually performed once per intra-period in conventional video encoders, we assume that the congestion control module determines the SBR R_S every intra-period [6]. In the meantime, packet loss process is monitored by the receiver, which keeps an estimate of the packet loss parameters and periodically feeds them back to the sender over the backward path $\mathcal{D} - \mathcal{S}$. When there is no packet loss, we have $f_d = f_e$, therefore the sender optimizes q and f_e such that the perceptual quality Q is maximized, under the sending rate constraint [2]. However, in the presence of random packet losses, f_d is a random variable that depends on the vector $\boldsymbol{m} = [m_1, \dots, m_N]^T$ of the number of redundant FEC packets for each frame, frame size vector $\boldsymbol{k} = [k_1, \dots, k_N]^T$, number L of temporal layers and finally the eFR f_e. Then, given the estimated loss parameters $\xi_{0|1}$ and $\xi_{1|0}$, along with the sending bitrate R_S, the sender is tasked with optimizing q, f_e and $\boldsymbol{m}$, so as to maximize the mean perceptual quality $\mathbb{E}(Q) \triangleq \overline{Q}$ of the decoded video for each new intra-period by solving the following problem.

$$
\begin{aligned}
\max_{q, f_e, \boldsymbol{m}} \quad & \overline{Q} = \mathrm{NQQ}(q) \cdot \overline{\mathrm{NQT}}(f_d(q, f_e, \boldsymbol{m})) \\
\text{s.t.} \quad & R(q, f_e) + \frac{B}{T} \sum_{i=1}^{N} m_i \leq R_{\mathrm{S}} \\
& q > 0,\ m_i \in \mathbb{N},\ f_e \in \mathcal{F}
\end{aligned}
\tag{2}
$$

Here, $\mathcal{F}$ is the set of frame rates considered, and $R(q, f_e)$ is the bitrate of the encoded video, which is estimated by the rate model in [9] as follows.

$$
R(q, f_e) = R_{\max} \left(\frac{q_{\min}}{q}\right)^{\beta_q} \left(\frac{f_e}{f_{\max}}\right)^{\beta_f}
\tag{3}
$$

Problem (2) is a mixed integer programming problem, and is hard to solve in general. Since $\mathcal{F}$ is a small set, optimizing f_e via exhaustive search in $\mathcal{F}$ is feasible. Then, for a particular value of f_e, the task is to optimize q and $\boldsymbol{m}$. Note that, even though q is continuous, the resulting frame size vectors $\{\boldsymbol{k}\}$ are discrete due to packetization. To predict distinct frame size vectors systematically, we assume that the frames in the same temporal layer have equal sizes, and develop a mean frame size prediction $\hat{z}_l$ per temporal layer l that depends on the target bitrate R linearly, given the eFR f_e, as well as the video contents. Accordingly, we choose to replace the optimization variable q with the video bitrate R, using Eq. (3). Next, we perform a hill climbing search in video bitrates $R \leq R_S$, however without using a predetermined step-size. Instead, we consider $R \leq R_S$ that result in distinct frame size vectors, which are estimated by the aforementioned model for a (f_e, R) tuple. In a bitrate interval $(R_0, R_1]$ that maps to the same prediction $\hat{\boldsymbol{k}} = \lceil \hat{\boldsymbol{z}}/B \rceil$, the highest bitrate R_1 leads to the maximum perceptual quality. The hill-climbing search is carried out over such suprema, starting from R_S and decreasing (line 8). At each step of hill climbing, $\mathrm{NQQ}\left(q(f_e, R)\right)$ is easily calculated from Eqs. (1) and (3) using the current R and the given f_e. Then, all that remains is to determine the FEC allocation vector $\boldsymbol{m}$ subject to the first constraint in Eq. (2) to maximize $\overline{\mathrm{NQT}}(f_d(q, f_e, \boldsymbol{m}))$. The entire procedure is summarized in Alg. 1. Note that, $\boldsymbol{m}$ is determined from its previous value for speed-up (line 11). In the next section, we will focus on this procedure, that is, how $\overline{\mathrm{NQT}}$ can be maximized.

IV. Determining the Frame-Level FEC Rates

Given the video bitrate R, we can create at most $M = \lfloor (R_S - R)T/B \rfloor$ FEC redundancy packets of size B to protect the frames in the intra-period. Let d_i denote the Bernoulli random variable that assumes the value 1 if the frame i is decoded at the receiver and 0 otherwise, and let $\mathbb{I}$ be the indicator function. Then, the total number of decoded frames is given by $D = \sum_{i=1}^{N} \mathbb{I}_{d_i = 1}$, and we can write

$$
0 \leq f_d = \frac{D}{T} \leq f_e = \frac{N}{T}.
$$

Algorithm 1 Frame Rate and Video Bitrate Opt.

```
*Inputs:* $\epsilon$, $\lambda$, $R_S$, $L$, $\mathcal{F}$, $T$, $B$, $\alpha_q$, $\alpha_f$, $\beta_q$, $\beta_f$, $f_{\max}$, $q_{\min}$, $R_{\max}$
*Outputs:* $f_e^*$, $R^*$, $\boldsymbol{m^*}$
$Q^* \leftarrow 0$
for all $f_e \in \mathcal{F}$ do                                              ▷ Exhaustive search
    $N \leftarrow T \times f_e$, $R \leftarrow R_S$, $Q \leftarrow 0$, $\boldsymbol{m} \leftarrow \boldsymbol{0}$, $\boldsymbol{k} \leftarrow \boldsymbol{0}$
    do                                                                        ▷ Hill-climbing begins
        $Q_{\text{max}} \leftarrow Q$, $R_{\text{best}} \leftarrow R$, $\boldsymbol{m}_{\text{best}} \leftarrow \boldsymbol{m}$
        $R \leftarrow \max_{0 \le R' \le R} R'$ s.t. $\hat{\boldsymbol{k}}(R') \neq \boldsymbol{k}$          ▷ Next $R$
        $\boldsymbol{k} \leftarrow \hat{\boldsymbol{k}}(R)$                                            ▷ Corresponding $\boldsymbol{k}$
        $M \leftarrow \lfloor (R_S - R)T/B \rfloor - \sum_{i=1}^{N} m_i$
        $\big(\boldsymbol{m}, \overline{\text{NQT}}\big) \leftarrow$greedyFEC$(M, \boldsymbol{k}, \boldsymbol{m}, L)$
        $Q \leftarrow \text{NQQ}(q(f_e, R)) \times \overline{\text{NQT}}$
    while $Q > Q_{\text{max}}$ and $R > 0$
    if $Q_{\text{max}} > Q^*$ then
        $f_e^* \leftarrow f_e$, $R^* \leftarrow R_{\text{best}}$, $\boldsymbol{m^*} \leftarrow \boldsymbol{m}_{\text{best}}$
    end if
end for
```

Ideally, we would like to optimize the FEC packet distribution $\boldsymbol{m}$, such that $\overline{\text{NQT}}(f_d)$ is maximized.

$$\begin{aligned} \max_{\boldsymbol{m}} \quad & \overline{\text{NQT}} = \sum_{n=0}^{N} \Pr(D = n) \text{NQT}(\frac{n}{T}) \\ \text{subject to} \quad & \sum_{i=1}^{N} m_i \le M \text{ and } 0 \le m_i \end{aligned} \tag{4}$$

Problem (4) aims to find the best way to distribute M redundancy packets on N frames, where the search space is $C(M + N - 1, M)$. Therefore, exhaustive search becomes infeasible even for small values of M. As a solution, we propose to use a greedy hill-climbing algorithm. We begin by assuming that only a single FEC packet is given, and increment the number of available FEC packets by one at each step. Next, we search among all the frames to determine which one should be protected with the newly added FEC packet. In other words, at step $n \le M$, we search among the neighbors of the best solution $\boldsymbol{m}^*$ found so far, where a neighbor $\boldsymbol{m'}$ of a vector $\boldsymbol{m}$ has the same components except for $m'_j = m_j + 1$. We can speed up this process by skipping the frames in the same layer for which the (k_i, m_i) tuple has already been evaluated. For each neighbor, we calculate a score; $\overline{\text{NQT}}$ in case of iid packet losses or $\overline{f_d}$ for Markovian packet losses, and then pick the neighbor that leads to the highest value as the new solution candidate. The reason for the choice of a different search space for the Markovian packet losses is the additional computational complexity. The algorithm, which is summarized in Alg. 2, terminates in $O(NM)$ iterations.

Algorithm 2 greedyFEC - FEC packet distribution

```
*Inputs:* $M$, $\boldsymbol{k}$, $\boldsymbol{m}$, $L$
*Outputs:* $\boldsymbol{m}$, $\overline{\text{NQT}}$
$\boldsymbol{m}^* \leftarrow \boldsymbol{m}$, $\overline{\text{NQT}}^* \leftarrow \overline{\text{NQT}}(\boldsymbol{m}^*)$                   ▷ initial solution
while $M > 0$ do                                                  ▷ distribute $M$ FEC packets
    $\boldsymbol{h} \leftarrow \boldsymbol{m}^*$
    for $1 \le i \le N$ do                                         ▷ over $N$ frames
        $\boldsymbol{m} \leftarrow \boldsymbol{h}$ and $m_i \leftarrow h_i + 1$
        if $\overline{\text{NQT}}(\boldsymbol{m}) > \overline{\text{NQT}}^*$ then
            $\overline{\text{NQT}}^* \leftarrow \overline{\text{NQT}}(\boldsymbol{m})$
            $\boldsymbol{m}^* \leftarrow \boldsymbol{m}$
        end if
    end for
    $M \leftarrow M - 1$
end while
```

Next, we show how to calculate $\overline{\text{NQT}}$ and $\overline{f_d}$ for iid and Markovian packet losses, respectively.

A. Independent, Identically Distributed Packet Losses

Our goal in this section is to calculate $\overline{\text{NQT}}$, given the FEC packet distribution $\boldsymbol{m}$ and that the packet losses are iid. From Eq. (4), this calculation is straight-forward once the probability distribution $P_D(n)$ of the number D of decoded frames is known, for each $n \in \{0, 1, \ldots, N\}$. In turn, $\Pr(D = n)$ can only be calculated by considering all possible decoding patterns in the intra-period that result in the decoding of exactly n frames. Note that, each of these decoding patterns corresponds to a particular sub-tree of the frame dependency tree $\mathcal{T}$, having n nodes, and with the I-frame as the root node.

When the packet losses are independent Bernoulli events with probability ϵ, frame arrivals are also independent, but with non-identical distributions due to the unequal error protection. Let a_i denote the Bernoulli random variable taking on the value 1 if the encoded frame i successfully arrives at the receiver. Note that d_i and a_i are different events, since an arriving frame cannot be decoded without its reference frame. Then, the arrival probability for frame i is

$$\Pr(a_i = 1) = \sum_{j=0}^{m_i} \binom{k_i + m_i}{j} \epsilon^j (1 - \epsilon)^{k_i + m_i - j},$$

that is, the probability of having at most m_i packet losses for frame i. We now make the following observation. If we let D_i be the number of decoded frames in the sub-tree $\mathcal{T}_i$ of the coding structure $\mathcal{T}$ rooted at frame i, the distribution of D_i is given by

$$\Pr(D_i = n) = \begin{cases} \Pr(d_i = 0) & n = 0 \\ \Pr(d_i = 1) \Pr(D_i' = n - 1), & n > 0 \end{cases} \tag{5}$$

where D_i' is the total number of frames in $\mathcal{T}_i$ decoded *given* frame i is decoded, which is in turn the summation of the number of decoded frames for each sub-tree of node i. This means that, due to the independence of the frame arrival events, the distribution of D_i' is given by the convolution of the D_j distributions of each child j of node i. Finally, if frame i belongs to the highest temporal layer L, we have $\Pr(D_i = n) = \Pr(a_i = n)$. Since $D_1 = D$ by definition, $\Pr(D = n)$ can be found recursively from Eq. (5) and taking convolutions. These steps are summarized in Alg. 3.

Algorithm 3 calcPMF (for iid loss)

```
*Inputs:* Frame index $i, \Pr(a_j = 1)$ for $1 \le j \le N$
*Output:* $P_{D_i}$
**if** $i \in \text{TL}(L)$ **then**                                   ▷ *Highest layer*
    $P_{D_i} = [1 - \Pr(a_i = 1), \Pr(a_i = 1)]$
**else**                                                              ▷ *Lower layer*
    $c = 0$
    **for all** $j$ predicted from $i$ **do**
        **if** $c = 0$ **then** $c = \text{calcPMF}(j)$
        **else** $c = c \ast \text{calcPMF}(j)$                        ▷ *convolution*
        **end if**
    **end for**
    $P_{D_i} = [1 - \Pr(a_i = 1), \Pr(a_i = 0) \times c]$
**end if**
```

B. Markovian Packet Losses

When the packet losses are Markovian and hence bursty, the frame arrivals become dependent, preventing us from using Alg. 3 to determine the end-to-end perceptual quality. Instead, we have to go through each possible frame decoding pattern, and calculate the corresponding probability of occurrence. As mentioned in Sec. IV-A, this is equivalent to enumerating all sub-trees of the frame dependency tree $\mathcal{T}$ with the I-frame as the root node, and becomes computationally cumbersome [2] for practical values of N and L. We circumvent this problem by maximizing the mean number of decoded frames $\overline{f_d}$ instead of $\overline{\text{NQT}}$. We have

$$\overline{f_d} = \frac{\mathbb{E}(D)}{T} = \frac{1}{T} \sum_{i=1}^{N} \mathbb{E}(\mathbb{I}_{d_i = 1}) = \frac{1}{T} \sum_{i=1}^{N} \Pr(d_i = 1). \tag{6}$$

[2] When $N = 32$, there are 2.015×10^6 and 3.187×10^6 sub-trees for 3 and 4 temporal layers, respectively.

Then, the decoding probability for frame i can be expanded as the multiplication of conditional arrival probabilities.

$$\begin{aligned}\Pr(d_i = 1) &= \prod_{j \in \mathcal{A}_i} \Pr(a_j = 1 | a_u = 1, \forall u \in A_j) \\ &\triangleq \prod_{j \in \mathcal{A}_i} p(j)\end{aligned} \tag{7}$$

Here, $\mathcal{A}_i$ is the set of ancestors of frame i in the coding structure. The conditional frame arrival probability $p(j)$ can be expanded [20] by further conditioning on the arrival state of frame j's first packet, which, in turn, depends on the arrival state of its reference frame's last packet. Towards this end, let $a_j^{(n)}$ denote the arrival state of the n^{th} packet that belongs to frame j. In particular, we consider the conditional probability $p_s(j)$ that the frame j arrives *and* the arrival state of its last packet is s, given that all the ancestors of frame j have arrived. Then, $p_s(j)$ is given by the following.

$$\begin{aligned}p_s(j) = \pi_0(j)\mathbb{I}_{m_j-2+s\geq 0} \sum\nolimits_{\ell=0}^{m_j-2+s} L_s(\ell, k_j + m_j - 2) \\ + \pi_1(j) \sum_{\ell=0}^{m_j-1+s} R_s(k_j + m_j - 2 - \ell, k_j + m_j - 2).\end{aligned} \tag{8}$$

In Eq. (8), we first condition on $a_j^{(1)}$ using the probability $\pi_s(j)$.

$$\pi_s(j) = \Pr(a_j^{(1)} = s \mid a_u = 1, \forall u \in \mathcal{A}_j) \tag{9}$$

Additionally, $R_s(u, v)$ and $L_s(u, v)$ denote the probabilities that there will be *exactly* u received or lost packets in the next $v \geq u$ packets, which are then followed by a packet in state s, given that the first packet is received or lost, respectively. The upper-triangle matrices R_0, R_1, L_0 and L_1 can be pre-computed easily via memoization [20], starting from the base case $(0, 0)$. The final step is to calculate $\pi_s(j)$. Let $\mathcal{P}_j$ be the reference frame from which frame j is predicted. Then it holds that

$$\begin{bmatrix}\pi_0(i) \\ \pi_1(i)\end{bmatrix} = \begin{bmatrix}1 - \xi_{1|0} & \xi_{0|1} \\ \xi_{1|0} & 1 - \xi_{0|1}\end{bmatrix}^{\kappa+1} \begin{bmatrix}p_0(\mathcal{P}_j) \\ p_1(\mathcal{P}_j)\end{bmatrix}, \tag{10}$$

where κ is the number of packets between frame $\mathcal{P}_j$'s last packet and frame j's first. Starting from frame 1 up to frame N, we can calculate $p_s(j)$ for all frame $1 \leq j \leq N$ using Eqs. (8), (9) and (10). Then, $p(j)$ is simply given by $p_0(j) + p_1(j)$.

V. Evaluations

In this section, we evaluate the performance of the proposed system. We present the maximized mean end-to-end Q-STAR value $\overline{Q}$ and the maximizer eFR f_e and the FEC bitrate ratio $r \triangleq 1 - R/R_S$ determined by our scheme for different video sequences, subject to iid and bursty packet losses, considering both hPP and IPP structures. In each evaluation, 10-second-long video sequences "Crew", "City", "Harbour" and "Soccer" are tested [21], while R_S is varied from 100 kbps to 1.6 Mbps with 30 kbps stepsizes. Due to space constraints, we present only the results obtained with "Harbour" and "City" in most cases, please see [22] for all results. For hPP, we also examine the mean FEC redundancy ratio $r(l)$ within layer l, given by $r(l) \triangleq \frac{1}{n_l} \sum_{i \in \mathrm{TL}(l)} r_i$, where n_l is the number P-frames that belong to layer l in the intra-period. Clearly, $1 + \sum_{l=1}^{L} n_l = N$.

Encoder settings: We use the x264 encoder with "High" profile, "very fast" preset and tuned for "zero latency" [23], and choose its one-pass ABR rate control algorithm, suitable for ultra-low delay scenarios. For each video, the picture resolution is 4CIF, and the frame rate is selected from $\mathcal{F} = \{15\,\text{Hz}, 30\,\text{Hz}\}$. To generate a particular hPP structure, we modified the x264 encoder by altering the reference frames used before each frame encoding [22] according to the H.264/AVC standard [24]. In our modification, the GoP length is set to $G = 4$, leading to $L = 3$ temporal layers as shown in Fig. 2. QP-cascading is performed during encoding. An I-frame is inserted every $T = 16/15$ seconds, meaning there are $N = 32$ and $N = 16$ frames in an intra-period for $f_e = 30\,\text{Hz}$ and $f_e = 15\,\text{Hz}$, respectively. To generate an IPP structure, x264 is used without any modification.

Q-STAR and R-STAR parameters: As described above, we have $f_{\max}$ = 30 Hz and $R_{\max}$ = 1.6 Mbps. The Q-STAR parameters α_q and α_f are taken from [1]. To derive the R-STAR parameters β_q and β_f for either IPP or hPP, we encode each video sequence with the corresponding encoder at target bitrates varied from 100 kbps to 1.6 Mbps with 30 kbps stepsizes, at each $f_e \in \mathcal{F}$. At each target bitrate, the average QP in each frame is obtained and converted[3] to QS, which are then averaged over all frames to determine the mean q. Using these q values, the actual video bitrates R and the f_e used, we derive β_q and β_f via curve-fitting with respect to Eq. 3. Finally, the parameter $q_{\min}$, which is used to evaluate *both* IPP and hPP for fair comparison, corresponds to the minimal QS obtained with the IPP structure at $R = R_{\max}$ and $f_e = 30$ Hz. Parameter values are listed in Table I. The corresponding $Q(R)$ curves for both hPP and IPP can be seen in Fig. 3.

[3] In H.264, QP $= 4 + 6\log_2(q)$.

TABLE I. QSTAR AND RSTAR PARAMETERS FOR IPP AND HPP WITH 3 TLS. X264 SETTINGS: "HIGH" PROFILE, "VERY FAST" PRESET, "ZERO LATENCY" TUNING. $f_{\max} = 30\,\text{Hz}$.

	α_q	α_f	β_q^{hPP}	β_f^{hPP}	β_q^{IPP}	β_f^{IPP}	$q_{\min}$
Crew	4.51	3.09	1.061	0.707	1.064	0.662	22.271
City	7.25	4.10	1.142	0.471	1.247	0.449	18.206
Harbour	9.65	2.83	1.320	0.584	1.461	0.489	34.301
Soccer	9.31	2.23	1.194	0.598	1.196	0.579	20.019

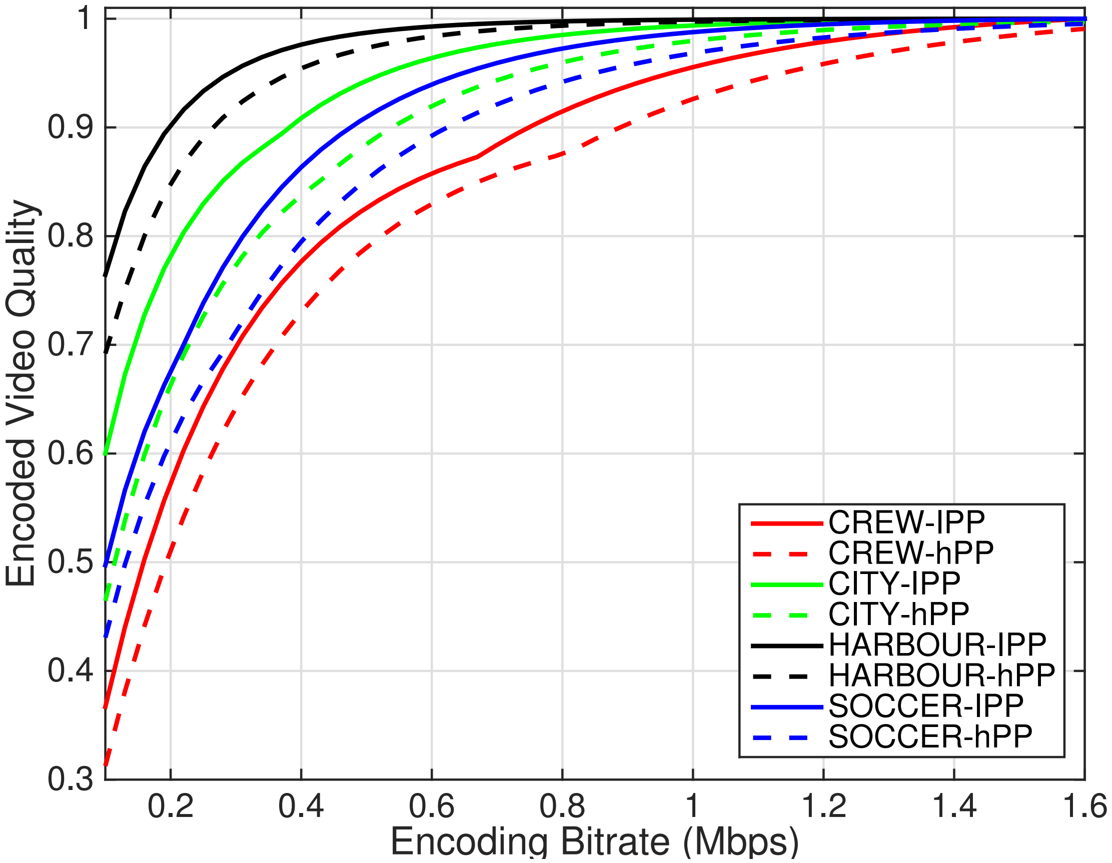

Fig. 3. Normalized Q-STAR perceptual quality models of video sequences with respect to the bitrate, using IPP (bold) and hPP (dashed) structures.

Modeling the Frame Sizes: As mentioned in Section III, we develop a model for the average frame size $\hat{z}(l, R)$ in each temporal layer l, which depends on the encoding bitrate R. Towards this end, we make use of the encoded video sequences described above. At each target bitrate R, we normalize the size z_i of each P-frame i with respect to the size z_0 of the I-frame within the same intra-period, and find the average normalized frame size $\tilde{z}(l, R)$ in each temporal layer l over the full duration of the video. The mean and the coefficient of variation of $\tilde{z}(l, R)$ in each layer l over the target bitrate range $R \in [100, 1600]$ kbps is given in Table II. We can see that, for each video sequence, the normalized mean frame size $\tilde{z}(l)$ in layer l shows little variation with the target bitrate, which means that it can be modeled independent of R.

Next in Fig. 4, we plot $\tilde{z}(l)$ against the temporal prediction distance τ_l within each layer l. By examining the general trend of how $\tilde{z}(l)$ changes with the temporal prediction distance τ_l, we propose the following model.

$$\tilde{z}(l) = 1 - e^{-\theta \tau_l^{\eta}} \tag{11}$$

The parameters θ and η can be found by curve-fitting in Figure 4. Once the normalized mean frame size $\tilde{z}(l)$ is predicted, we can estimate the actual size $\hat{z}(l, R)$ of a layer-l P-frame at a particular bitrate R by distributing the total number of bits RT in the intra-period among the frames proportional to $\tilde{z}(l)$.

$$\hat{z}(l, R) = RT \frac{\tilde{z}(l)}{1 + \sum_{l=1}^{L} n_l \tilde{z}(l)} \tag{12}$$

TABLE II. MEAN / COEFFICIENTS OF VARIATION OF $\tilde{z}(l)$ FOR $l \in \{1, 2, 3\}$, OVER $R \in [100, 1600]$ KBPS

$f_e = 30$ Hz	**TL(1)**	**TL(2)**	**TL(3)**
Crew	0.559 / 0.037	0.451 / 0.035	0.361 / 0.031
City	0.444 / 0.063	0.382 / 0.057	0.299 / 0.057
Harbour	0.462 / 0.079	0.402 / 0.100	0.300 / 0.111
Soccer	0.508 / 0.101	0.426 / 0.109	0.326 / 0.098
$f_e = 15$ Hz	**TL(1)**	**TL(2)**	**TL(3)**
Crew	0.815 / 0.030	0.733 / 0.025	0.611 / 0.028
City	0.670 / 0.049	0.594 / 0.061	0.508 / 0.057
Harbour	0.730 / 0.036	0.641 / 0.023	0.543 / 0.026
Soccer	0.802 / 0.029	0.733 / 0.039	0.549 / 0.061

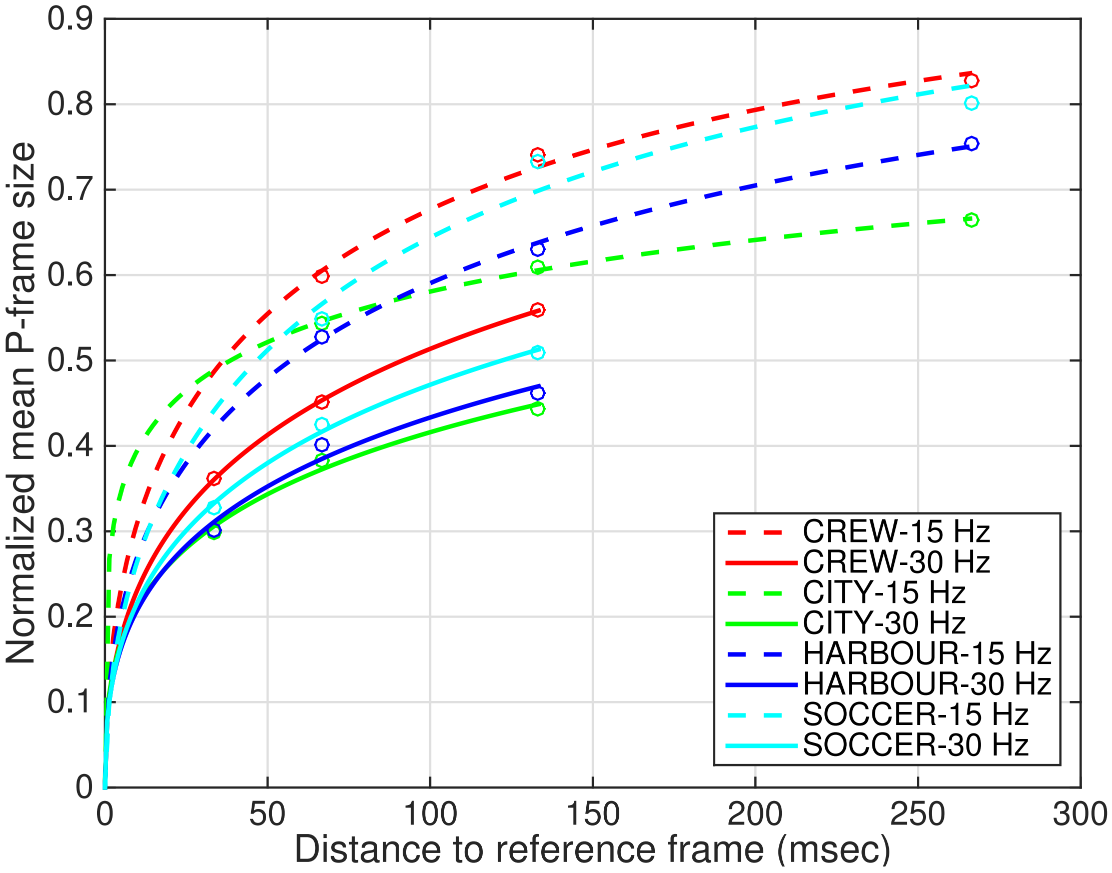

Fig. 4. Normalized mean frame sizes (circles) and the corresponding fits (bold and dashed lines) with respect to the temporal distance to their reference frames. $L = 3$ and for $f_e = 15$, $\tau_1 = 266.6$ ms, while for $f_e = 30$, $\tau_1 = 133.3$ ms. We have $\tau_1 = 2\tau_2 = 4\tau_3$.

The frame sizes in units of packets is given by $\hat{k}(l, R) = \lceil \hat{z}(l, R)/B \rceil$. In our evaluations, the payload size was chosen to be $B = 200$ bytes.

A. Evaluations for iid Packet Losses

We begin our evaluations with iid losses, for which we set $\xi_{0|1} + \xi_{1|0} = 1$. Packet loss rate ϵ is varied from 0.05 to 0.2.

1) Using the hPP structure: In Fig. 6, the Q-STAR scores $\overline{Q}_{\mathrm{hPP}}$ achieved with the hPP structure are shown for each video. We can see that the achieved $\overline{Q}_{\mathrm{hPP}}$ drops with the increasing ϵ as expected, but this drop is kept minimal by FEC. For a given video, the FEC bitrate ratio $r_{\mathrm{hPP}}(\epsilon)$ at packet loss rate ϵ is similar regardless of R_S as seen in Fig. 7, and only fluctuates due to the discrete nature of the problem, which is a consequence of packetization. For Harbour, when $R_S \geq 790$ kbps, FEC is able to ensure that the $\overline{Q}_{\mathrm{hPP}}$ is at least 96% of the lossless quality, since the perceptual quality model for this sequence is not largely sensitive to increases in QS, but to decreases in dFR. As a result, r_{hPP} is higher for Harbour, trading off the less important amplitude resolution with the more important end-to-end temporal resolution. In Table III, we can see that the preferred eFR jumps from 15 Hz to 30 Hz at higher sending bitrates for higher packet loss rates. For Harbour, this jump happens at a low

TABLE III. SENDING BITRATE (MBPS) AT WHICH PREFERRED f_e TRANSITIONS FROM 15 HZ TO 30 HZ, CASE OF IID PACKET LOSSES

hPP	$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.15$	$\epsilon = 0.2$
Crew	0.82	1	1.09	1.18	1.3
City	0.46	0.52	0.55	0.58	0.61
Harbour	0.13	0.16	0.22	0.22	0.25
Soccer	0.31	0.4	0.46	0.52	0.55

IPP	$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.15$	$\epsilon = 0.2$
Crew	0.67	0.94	1.03	1.12	1.21
City	0.37	0.55	0.64	0.7	0.73
Harbour	≤0.1	0.16	0.22	0.25	0.28
Soccer	0.22	0.4	0.46	0.49	0.55

TABLE IV. AFFINE MODEL PARAMETERS FOR FEC BITRATE RATIO (VALID FOR IID LOSSES)

	a_{hPP}	b_{hPP}	a_{IPP}	b_{IPP}
Crew	1.534	6.417	1.288	13.540
City	1.272	9.178	1.207	17.276
Harbour	1.274	15.689	1.320	22.339
Soccer	1.371	10.936	1.294	18.170

bitrate of $R_S \leq 300$ kbps for the reason mentioned before. In Figure 5, we show how the layer redundancy ratios $r(l)$ changes with R_S for each layer l. We can see that the redundancy ratios of higher layers are smaller, indicating unequal error protection. However, as R_S grows, the layer redundancy ratios all converge on the FEC bitrate ratio $r_{\text{hPP}}(\epsilon)$. Finally, when plotted against the packet loss rates in Fig. 8, we can see that the FEC bitrate ratio r can be modeled as an affine function of ϵ, for *both* hPP and IPP structures.

$$r(\epsilon) = a \cdot \epsilon + b$$

The corresponding model parameters are given in Table IV.

2) Using the IPP structure: IPP structure provides higher bitrate efficiency compared to hPP (Fig. 3), but lacks the resiliency offered by temporal layering. In terms of the achieved Q-STAR scores, we see that $\overline{Q}_{\text{IPP}} > \overline{Q}_{\text{hPP}}$ for all scenarios in Fig. 9. Further inspection reveals that the IPP structure delivers similar dFR as hPP, but at the cost of more FEC bits, i.e., $r_{\text{IPP}} > r_{\text{hPP}}$ (Fig. 8). Ultimately, even at smaller video bitrates, IPP can still have smaller QS due to hPP's relatively high coding overhead, and thus achieves higher $\overline{Q}$.

It may appear that the IPP structure outperforms hPP from the perceptual quality perspective. However, it is worth noting that, whenever a frame is lost, the rest of the frames in the intra-period is rendered undecodable in the IPP structure, whereas the hPP structure is able to continue the stream with the lower layer frames if the lost frame belongs to enhancement layers. To test this claim, we conduct 10^5 simulations with each coding structure and for each scenario, and we determine the sample mean $\hat{\mu}_\tau$ and the sample standard deviation $\hat{\sigma}_\tau$ of the time-averaged temporal distance τ between the decoded frames for both IPP and hPP[4]. We define τ, for a particular frame decoding pattern, as the weighted average of inter-frame distances, with weights equal to the time fraction that a particular inter-frame distance is observed. As an example, in Fig. 1, if P5, P6 and P7 were discarded, then the observed frame distance would be 1 unit in the first half of the intra-period, and 4 units in the rest, assuming the next I-frame is decoded. In this case, the time-averaged frame distance is $\tau = 0.5 \cdot 1 + 0.5 \cdot 4 = 2.5$ units. Among distinct frame decoding patterns with the same number of decoded frames, τ is minimized by having minimal frame distance variation.

The differences in the sample mean $\hat{\mu}_\tau^{\text{IPP}} - \hat{\mu}_\tau^{\text{hPP}}$ and the sample standard deviation $\hat{\sigma}_\tau^{\text{IPP}} - \hat{\sigma}_\tau^{\text{hPP}}$ in milliseconds can be seen in Figures 10 and 11. We can observe that, hPP presents smaller mean and variance in the frame intervals at sending bitrates up to 750 kbps; in other words, the decoded frames can be displayed with more regular intervals in this bitrate range. Ultimately, we conjecture that taking advantage of the temporal layering in hPP is advantageous for small sending bitrates. However, as the sending bitrate grows larger, FEC can be applied more efficiently, and temporal layering loses its edge for protection.

B. Evaluations for Bursty Packet Losses

For bursty packet losses, we focus on the effects of the average burst length λ at a constant packet loss rate. We set $\epsilon = 0.1$ and try $\lambda = \{2, 5, 10, 50\}$. To better investigate the layer redundancies and make a clear comparison between IPP and hPP, we use a constant $f_e = 15\,\text{Hz}$ in all our evaluations.

1) Using the hPP structure: Figure 12 shows that the Q-STAR scores $\overline{Q}_{\text{hPP}}$ achieved drop further as the average burst length λ increases. Unlike the case of iid losses, the FEC bitrate ratio r_{hPP} varies across the SBR range in Figure 13, where the horizontal axis is replaced with the average number of packets per FEC code block, which is proportional to SBR. Particularly for $\lambda = 50$, r_{hPP} is much smaller when $R_S \leq 600$ kbps (avg. FEC block size = 25 packets). Note that, increasing λ while keeping ϵ constant also increases the average "good" channel length, which equals $1/\xi_{0|1}$. This means that the probability of a packet loss within an intra-period diminishes, requiring less FEC protection. However, when bursts eventually happen, a large number of packets is lost, rendering multiple consecutive frames undecodable. Such bursty frame losses are severe at high λ and low SBR values, and they require infeasibly large r_{hPP} values. The growing of r_{hPP} eventually disappear at higher SBR values and each curve converges. Note that, the converged points grow with larger λ. Similarly, Figures 14 and 15 show that the gap between the layer redundancies get narrower with increasing SBR, and wider with longer bursts.

[4] μ_x and σ_x denote the mean and the standard deviation of the random variable x, respectively.

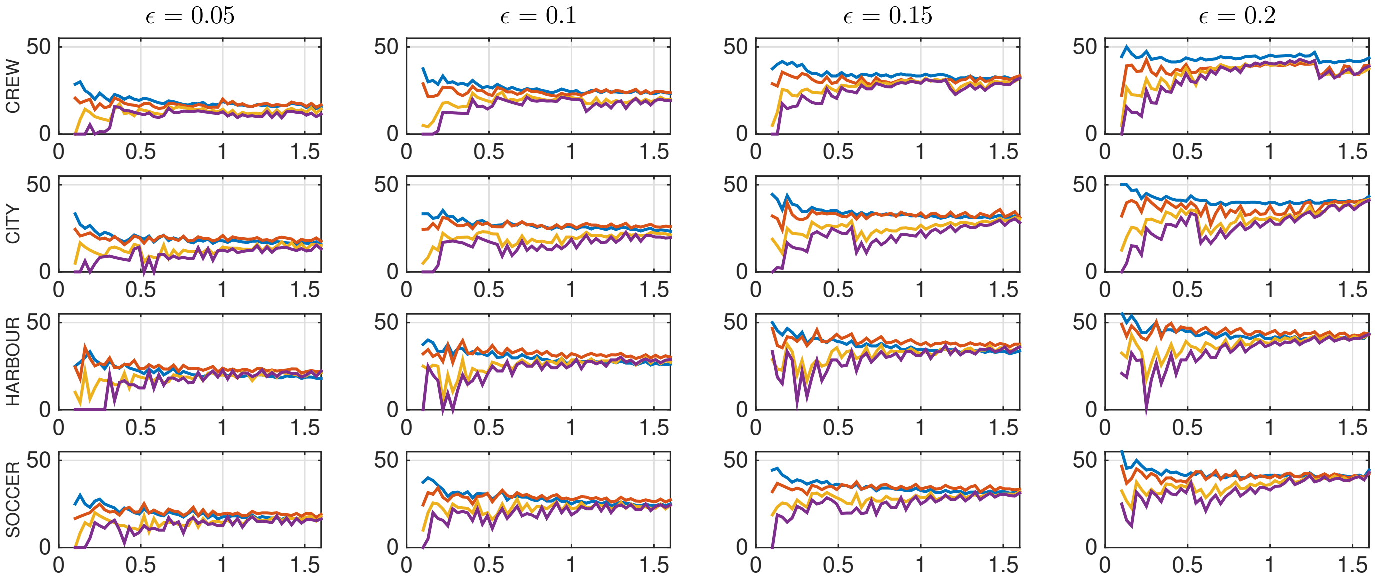

Fig. 5. Layer redundancy rates (%) for iid packet losses and using hPP structure with 3 TLs, showing I-frame (blue), TL(1) (red), TL(2) (yellow), and TL(3) (purple). Columns correspond to different ϵ values (see top), rows correspond to different video sequences (see left).

2) Using the IPP structure: Although we have $\overline{Q}_{\text{IPP}} > \overline{Q}_{\text{hPP}}$ with iid packet losses, Figure 16 shows that the IPP structure loses its advantage in all video sequences but City, when the packet losses are bursty with $\lambda \geq 5$. This is because, even though the hPP structure is prone to the coding overhead especially at low bitrates, its end-to-end frame rate is higher for bursty losses, leading to overall quality gains against IPP. In the same figure, the horizontal axis is replaced again with the average FEC code block size. We see that the relative Q-STAR gap between hPP and IPP is widest when the average burst length is about half the average frame size. This is because, if a frame is subject to such relatively long bursts and eventually gets lost, all other frames in the IPP structure are lost as well, while the decoding can be picked up, at the latest, from the next GoP if the hPP structure is used, provided the lost frame is not from the base layer. This hints at a disadvantage of the IPP structure when burst length is close to the average frame size. As the bursts get longer, hPP loses its effectiveness as well, since losing a base layer frame becomes more likely. Finally, we conduct 10^5 simulations again for each scenario to compare the decoded frame distance statistics for both IPP and hPP. We can observe from Figures 17 and 18 that, hPP presents significantly smaller mean and variance in the frame intervals in the whole SBR range, especially when $\lambda \geq 5$. Ultimately, we conjecture that using the hPP structure is advantageous for bursty losses with $\lambda \geq 5$.

VI. Conclusions

Real-time video delivery is prone to packet losses in the network. Achieving minimal latency and satisfactory QoE for such applications under high packet loss rate and long burst scenarios requires a joint optimization of video coding methods applied, video resolutions used, and frame-level FEC code rates on individual video frame. In this paper, leveraging on prior studies, we have used a perceptual video quality model that accounts for the effect of QS and the dFR separately, where dFR is simply approximated by the number of decodable frames per second. We explored methods to find the optimal video bitrate, the optimal FEC redundancy rate for each frame for the IPP and hPP structures, in addition to the trade-offs between choosing different frame rates and mean QS values that achieve a target video bitrate. Considering the finite-length FEC blocks, we formulated the quality maximization problem as a combinatorial optimization, and solved it using a combination of exhaustive search, hill-climbing and greedy methods. We have shown that, in case of iid losses up to 20%, the FEC bitrate percentage can be approximated as an affine function of the packet loss rate. Furthermore, IPP structure achieves higher perceptual quality values at all sending rates. This, however, does not necessarily mean that the actual perceptual quality with IPP is higher, because the IPP structure is more likely to lead to uneven frame intervals, reflected by longer mean frame intervals and higher variance of frame intervals at low sending rates. In case of bursty Markovian losses, we show that IPP structure loses its edge in providing a high quality, and therefore hPP is favored in such cases.

References

[1] Y.-F. Ou, Y. Xue, and Y. Wang, "Q-STAR: A Perceptual Video Quality Model Considering Impact of Spatial, Temporal, and Amplitude Resolutions," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2473–2486, 2014.

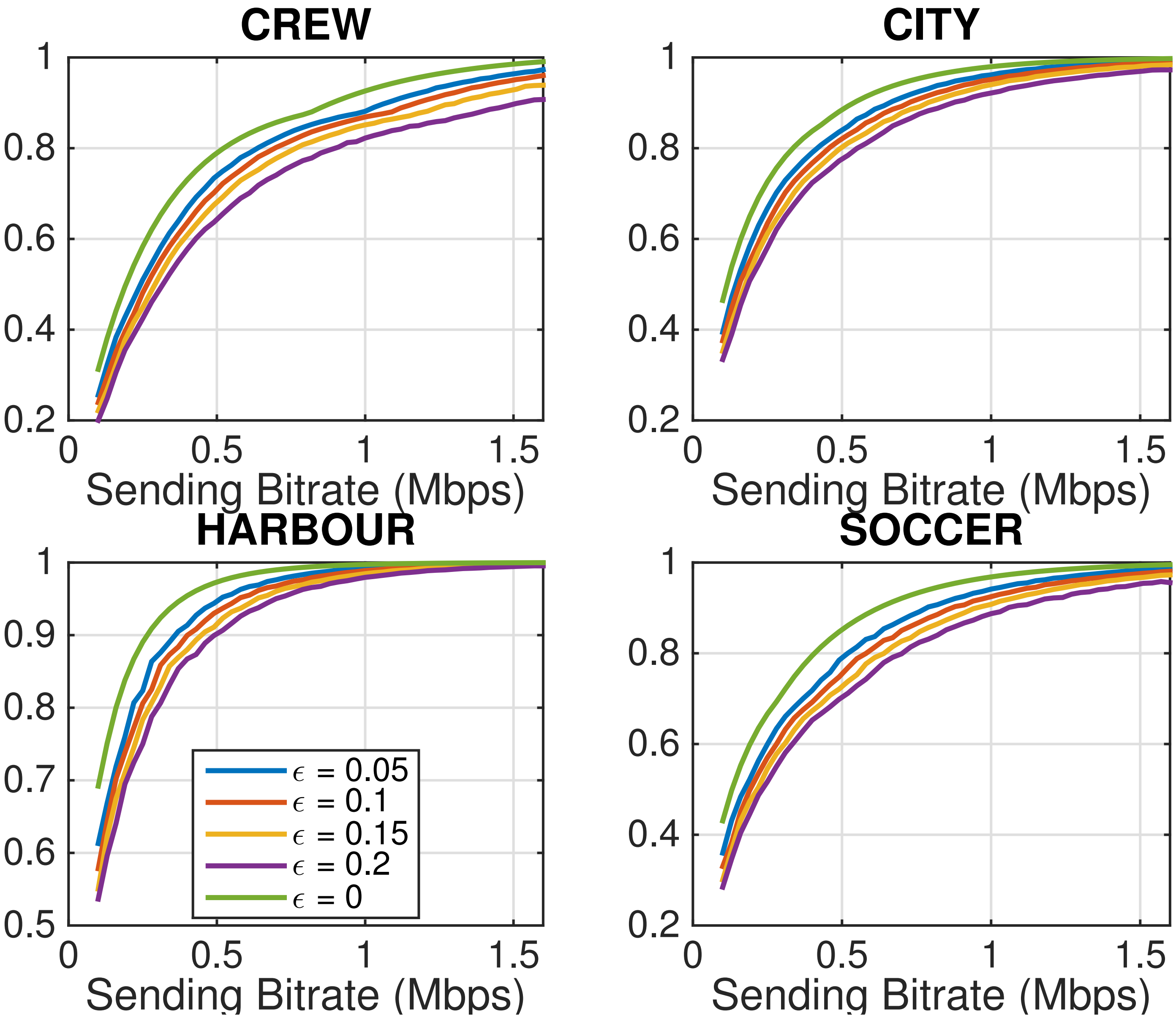

Fig. 6. Mean end-to-end Q-STAR scores achieved in case of iid packet losses, using hPP structure with 3 TLs.

[2] H. Hu, Z. Ma, and Y. Wang, "Optimization of Spatial, Temporal and Amplitude Resolution for Rate-Constrained Video Coding and Scalable Video Adaptation," in *Proc. of 19th IEEE International Conference on Image Processing*. IEEE, 2012.

[3] W. Chen, L. Ma, and C.-C. Shen, "Congestion-Aware MAC Layer Adaptation to Improve Video Telephony over Wi-Fi," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 5s, pp. 83, 2016.

[4] A. Bergkvist, D. C. Burnett, C. Jennings, and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers," *Working draft, W3C*, vol. 91, 2012.

[5] D. Hong, M. Horowitz, A. Eleftheriadis, and T. Wiegand, "H.264 Hierarchical-P Coding in the Context of Ultra-low Delay, Low Complexity Applications," in *Picture Coding Symposium (PCS), 2010*. IEEE, 2010, pp. 146–149.

[6] E. Kurdoglu, Y. Liu, Y. Wang, Y. Shi, C. Gu, and J. Lyu, "Real-time Bandwidth Prediction and Rate Adaptation for Video Calls over Cellular Networks," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016.

[7] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013.

[8] L. De Cicco, G. Carlucci, and S. Mascolo, "Experimental Investigation of the Google Congestion Control for Real-time Flows," in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. ACM, 2013.

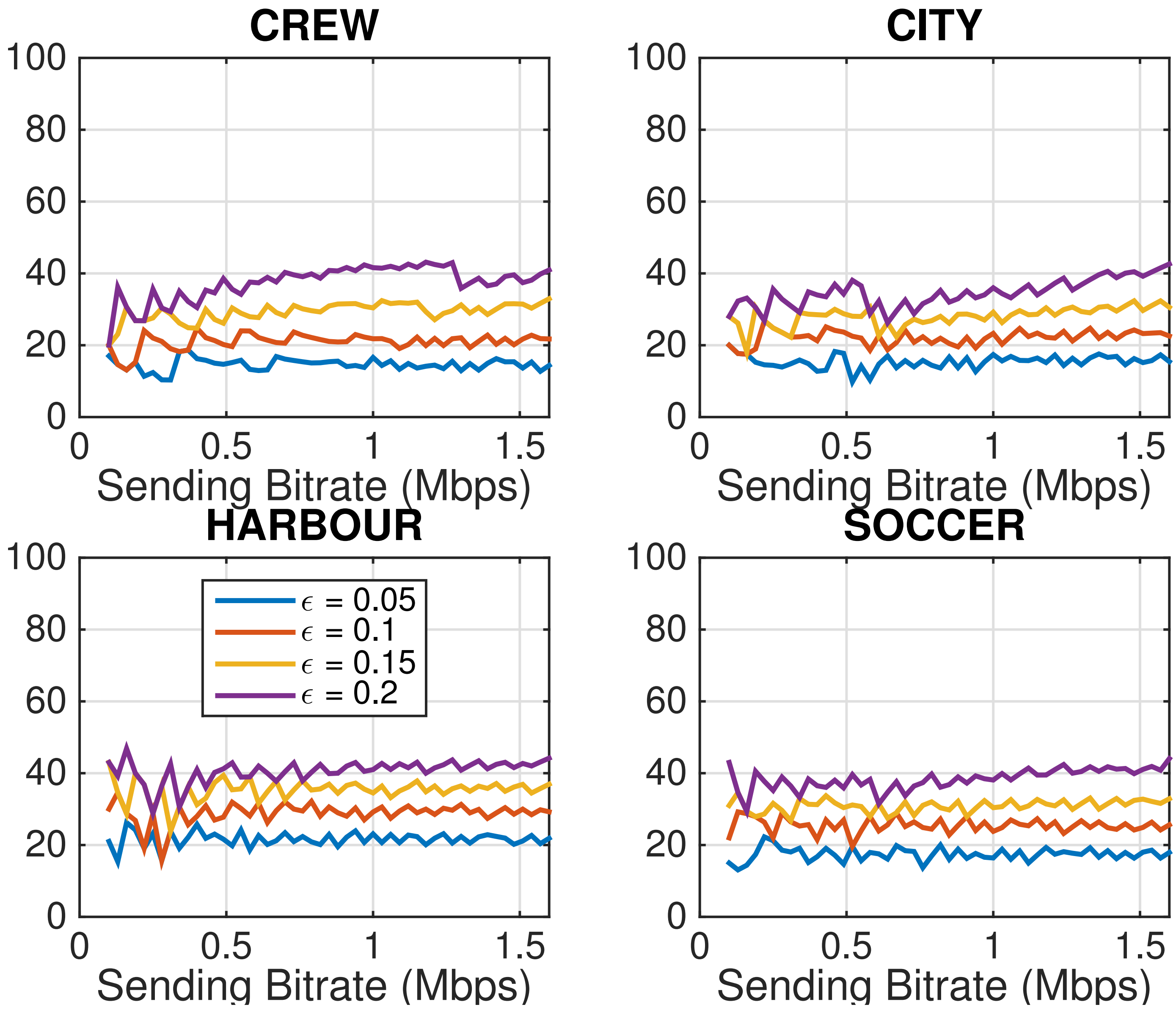

Fig. 7. FEC bitrate percentages r_{hPP} in case of iid packet losses, using hPP structure with 3 TLs.

[9] Z. Ma, F.C.A. Fernandes, and Y. Wang, "Analytical Rate Model for Compressed Video Considering Impacts of Spatial, Temporal and Amplitude Resolutions," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW),*. IEEE, 2013.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 2012.

[11] N. M. Freris, C.-H. Hsu, J. P. Singh, and X. Zhu, "Distortion-Aware Scalable Video Streaming to Multinetwork Clients," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 469–481, 2013.

[12] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of Video Transmission over Lossy Channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.

[13] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, "Streaming Mobile Cloud Gaming Video over TCP with Adaptive Source-FEC Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 10, 2016.

[14] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of Packet Loss for Compressed Video: Effect of Burst Losses and Correlation Between Error Frames," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 861–874, 2008.

[15] J. Xiao, T. Tillo, C. Lin, and Y. Zhao, "Dynamic Sub-GOP Forward Error Correction Code for Real-time Video Applications," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1298–1308, 2012.

[16] E. Maani and A. K. Katsaggelos, "Unequal Error Protection for Robust Streaming of Scalable Video over Packet Lossy Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, pp. 407–416, 2010.

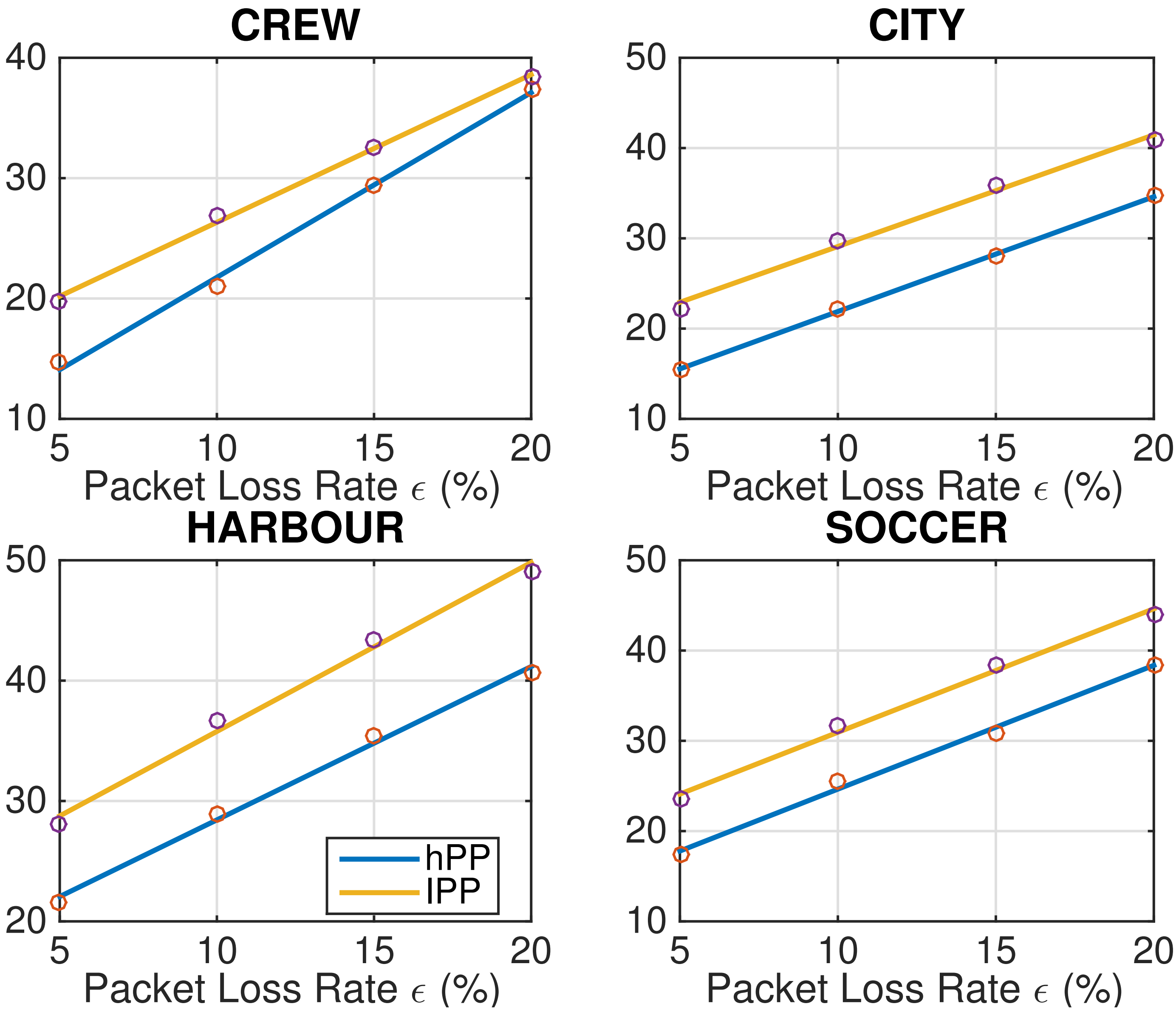

Fig. 8. FEC bitrate percentages r_{hPP} and r_{IPP} in case of iid packet losses, using hPP coding with 3 TLs and using IPP.

[17] H. Wu, M. Claypool, and R. Kinicki, "Adjusting Forward Error Correction with Temporal Scaling for TCP-friendly Streaming MPEG," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 1, no. 4, pp. 315–337, 2005.

[18] C.-H. Shih, C.-I Kuo, and Y.-K. Chou, "Frame-based Forward Error Correction Using Content-dependent Coding for Video Streaming Applications," *Computer Networks*, vol. 105, pp. 89–98, 2016.

[19] R. E. Blahut, *Theory and Practice of Error Control Codes*, vol. 126, Addison-Wesley Reading (Ma) etc., 1983.

[20] P. Frossard, "FEC Performance in Multimedia Streaming," *IEEE Communications Letters*, vol. 5, no. 3, pp. 122–124, 2001.

[21] C. Montgomery, "Xiph.org Video Test Media (derf's collection), https://media.xiph.org/video/derf," .

[22] NYU Video Lab, "Project Page, http://vision.poly.edu/index.html/index.php/HomePage/Fec4rt," .

[23] L. Merritt and R. Vanam, "x264: A High Performance H.264/AVC Encoder," 2006.

[24] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley & Sons, 2004.

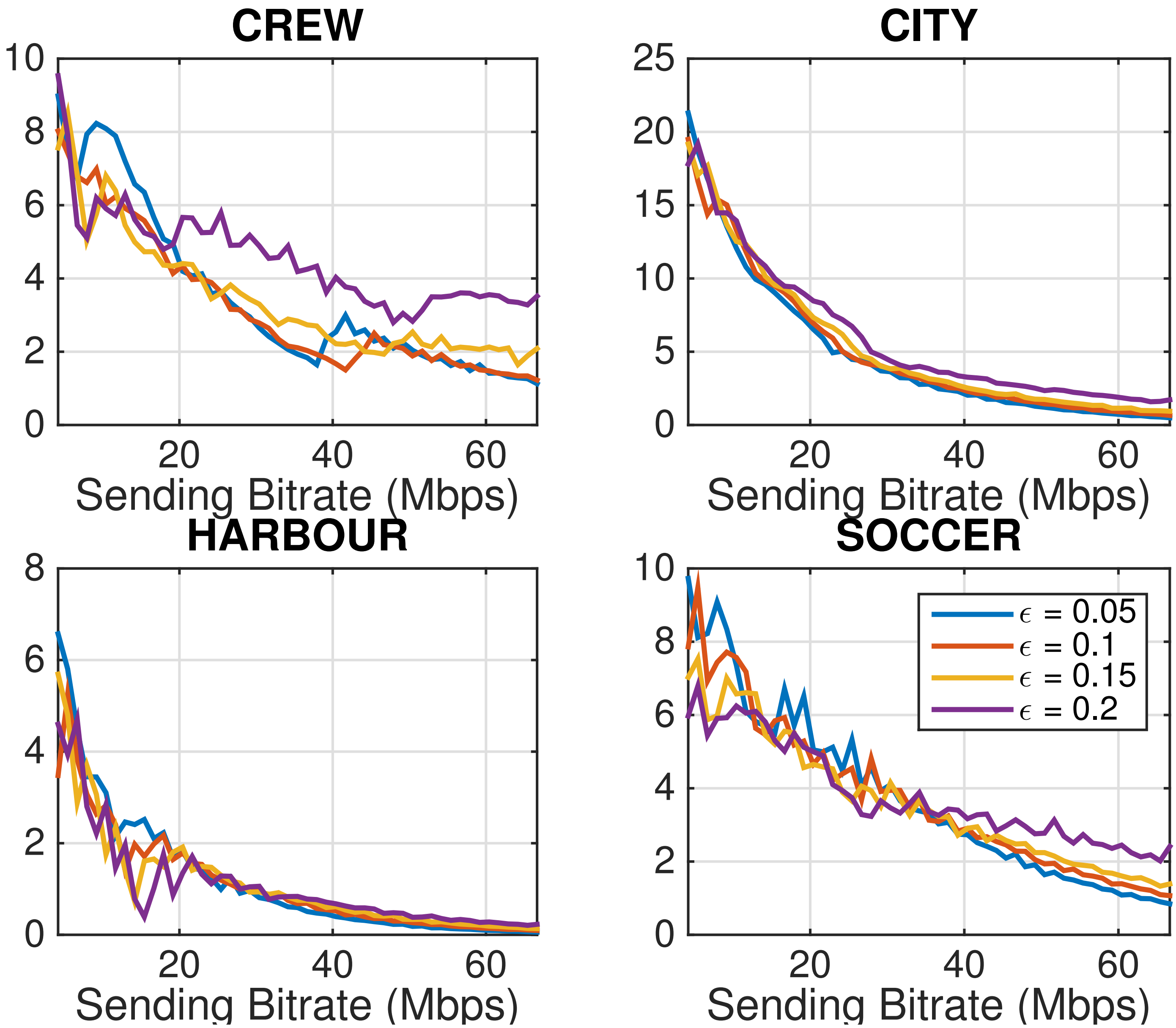

Fig. 9. Percentage of reduction in the achieved Q-STAR, relative to the IPP structure, given by $100 \times (1 - \overline{Q}_{\text{hPP}}/\overline{Q}_{\text{IPP}})$, in case of iid losses.

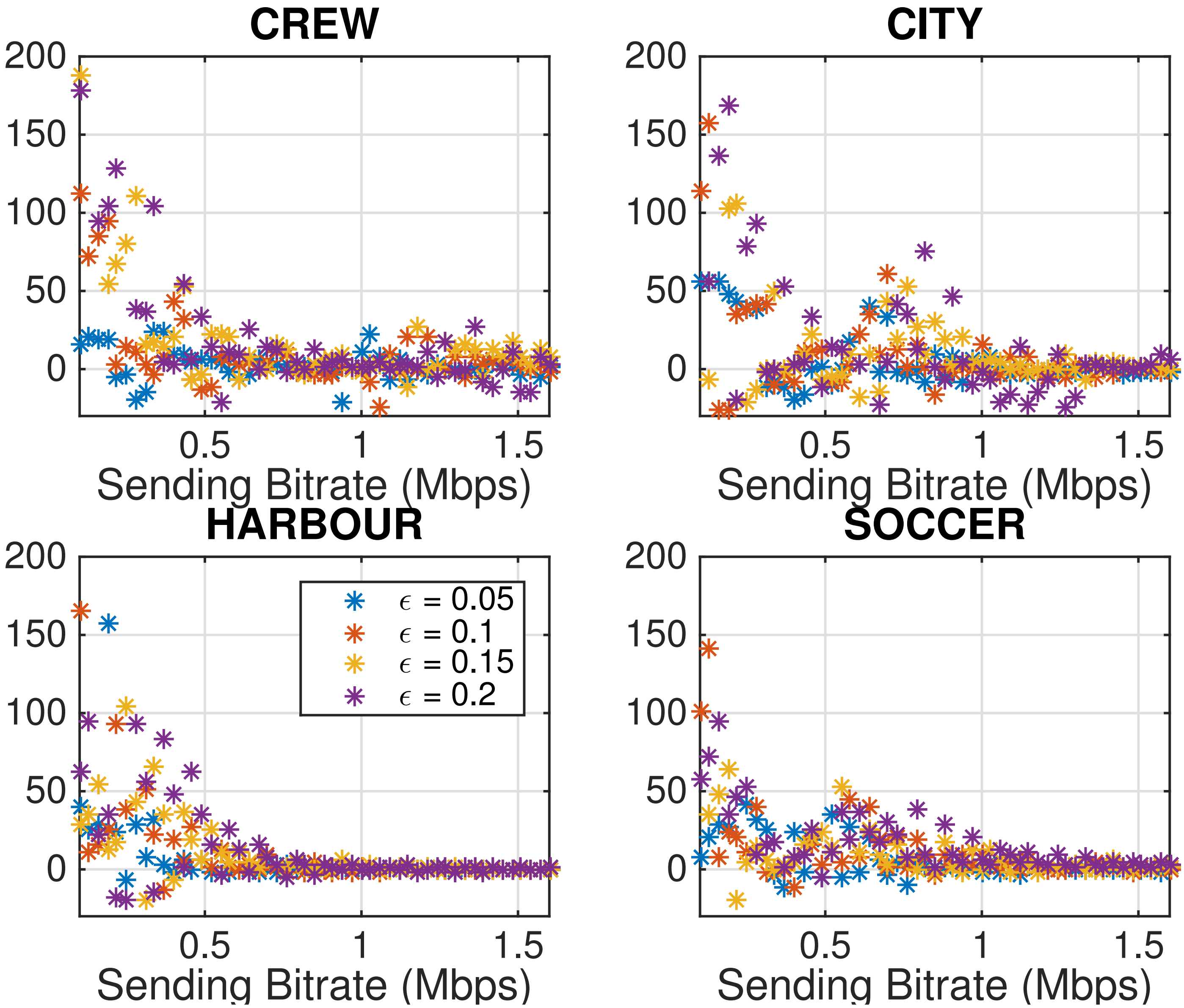

Fig. 10. Difference of mean frame intervals $\bar{\tau}_{\mathrm{IPP}} - \bar{\tau}_{\mathrm{hPP}}$ (msec) with IPP and hPP for iid packet losses, 10^5 simulations.

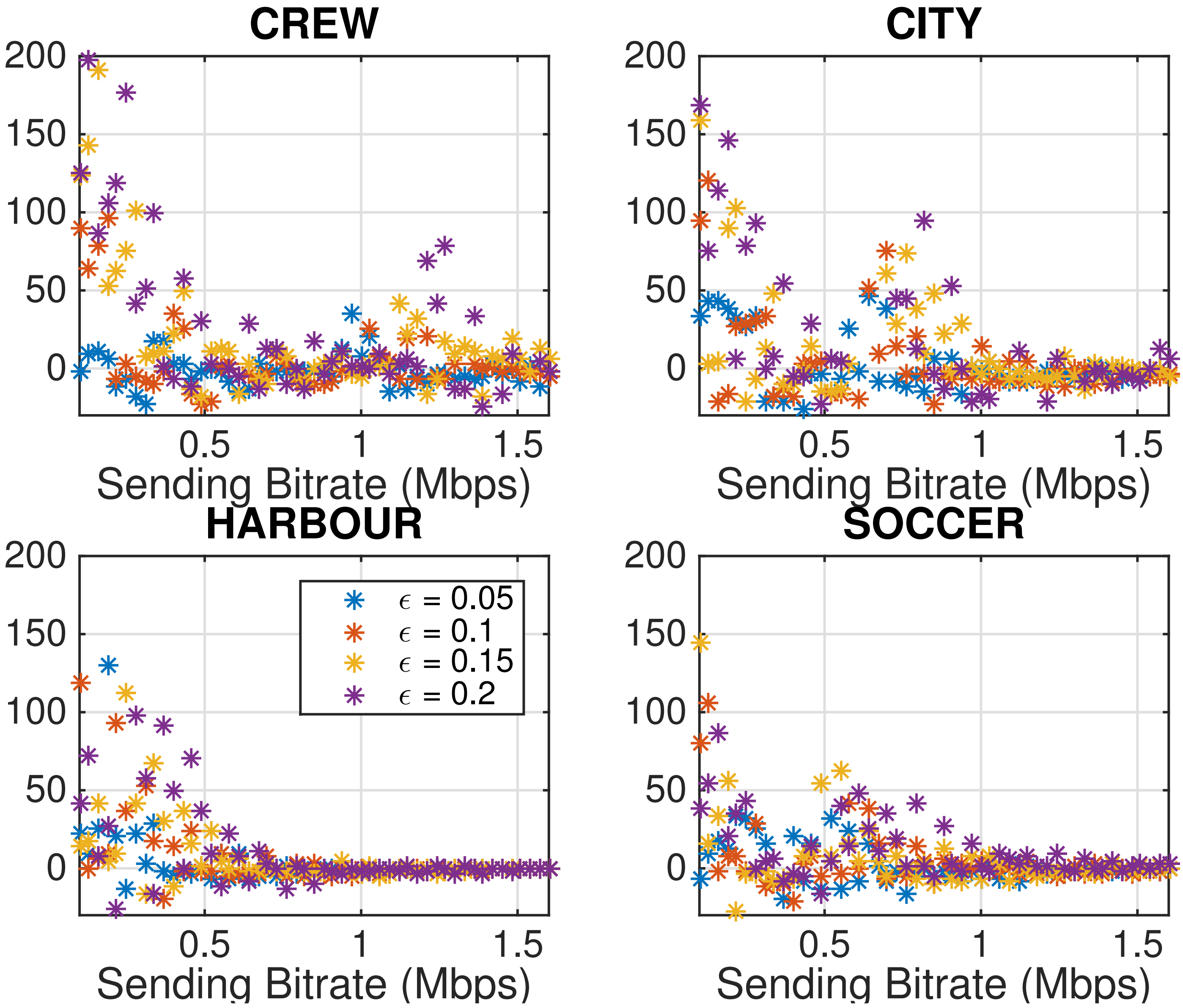

Fig. 11. Difference of standard deviations of frame intervals $\sigma_\tau^{\mathrm{IPP}} - \sigma_\tau^{\mathrm{hPP}}$ (msec) with IPP and hPP for iid packet losses, 10^5 simulations.

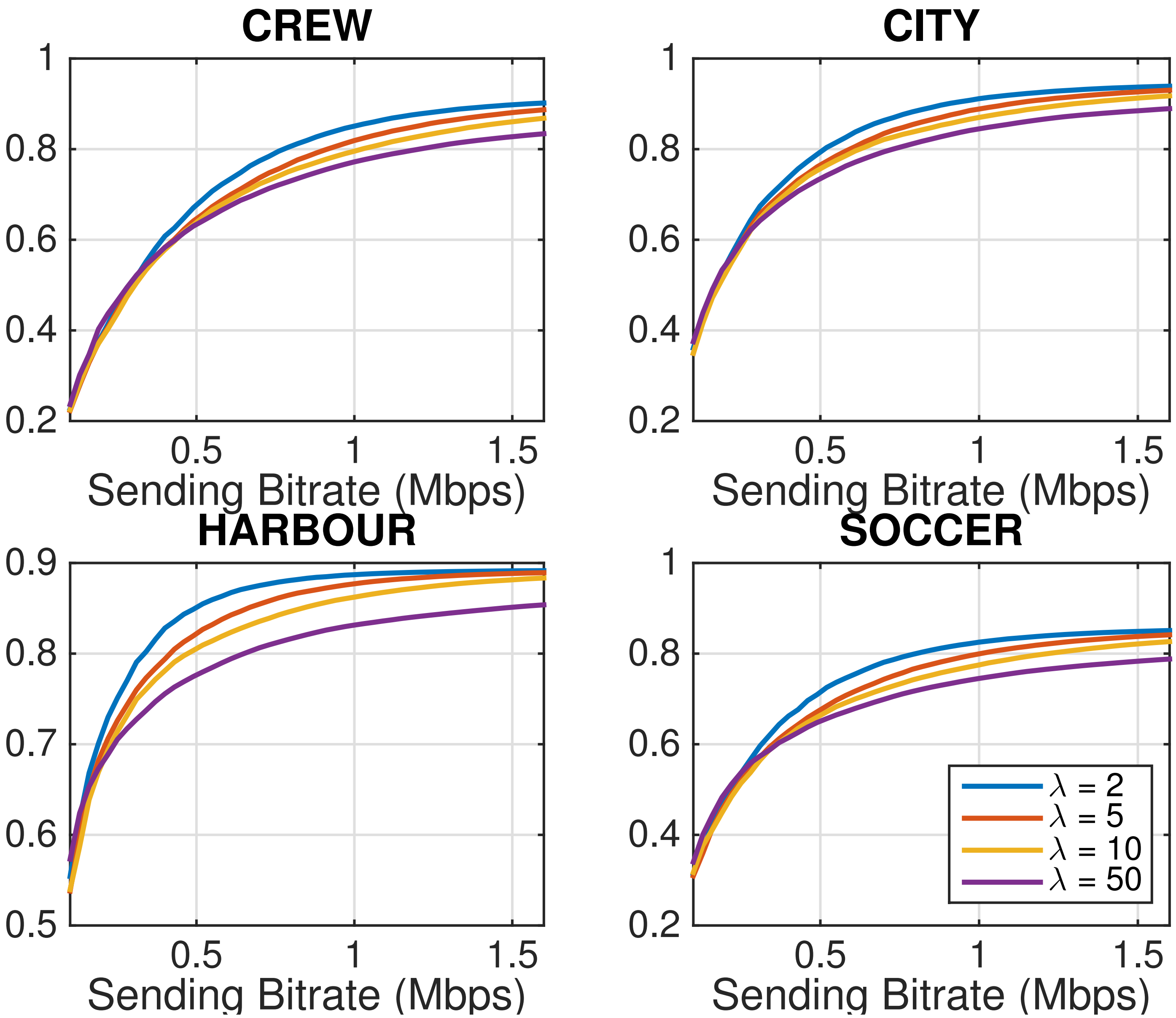

Fig. 12. Mean end-to-end Q-STAR scores achieved in case of bursty packet losses with $\epsilon = 0.1$, using hPP structure with 3 TLs and $f_e = 15\,\mathrm{Hz}$.

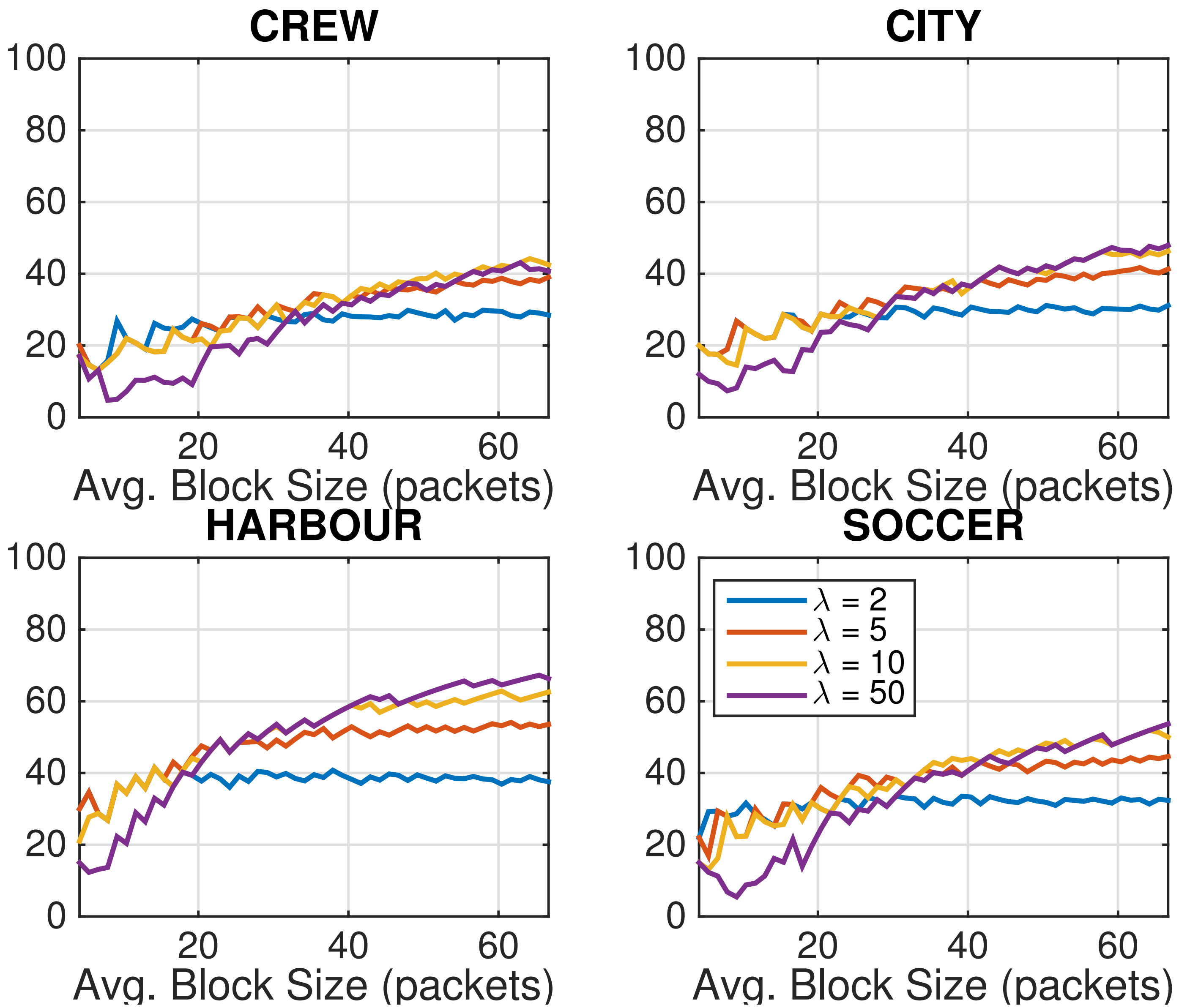

Fig. 13. FEC bitrate ratios in case of bursty packet losses with $\epsilon = 0.1$, using hPP structure with 3 TLs. Average FEC block size is calculated for each R_S, with $B = 200$ Bytes.

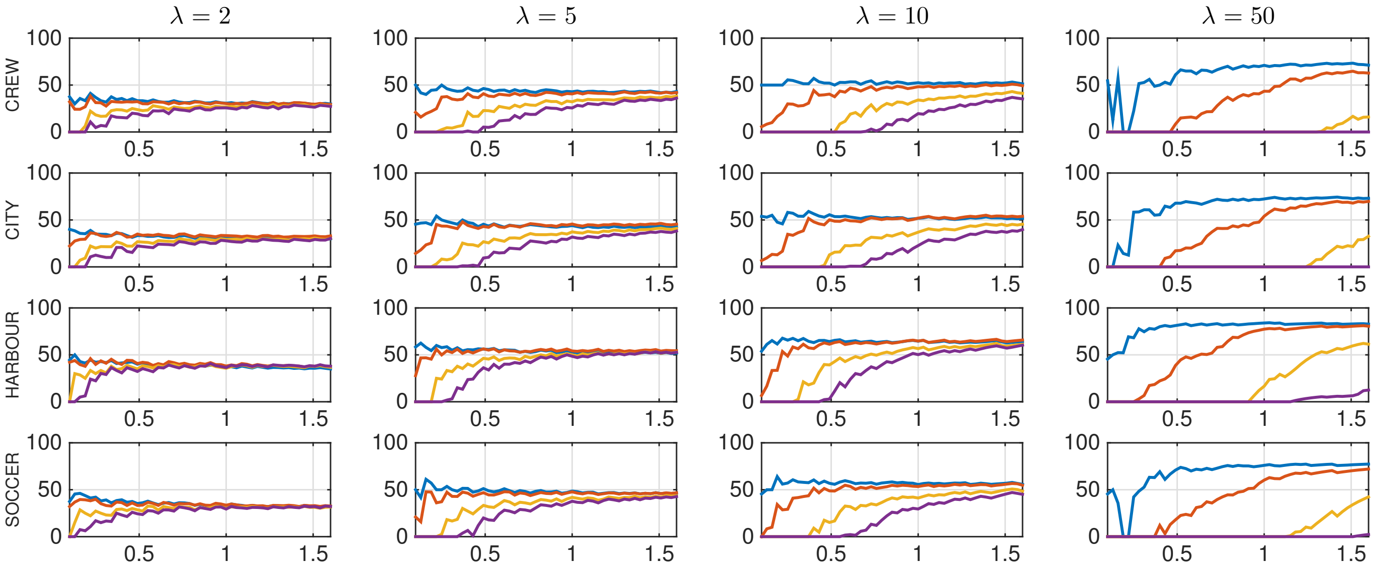

Fig. 14. Layer redundancy rates (%) for bursty packet losses and using hPP structure with 3 TLs, showing I-frame (blue), TL(1) (red), TL(2) (yellow), and TL(3) (purple). Horizontal axis is SBR (Mbps). Columns correspond to different λ values (top), rows correspond to different video sequences (left).

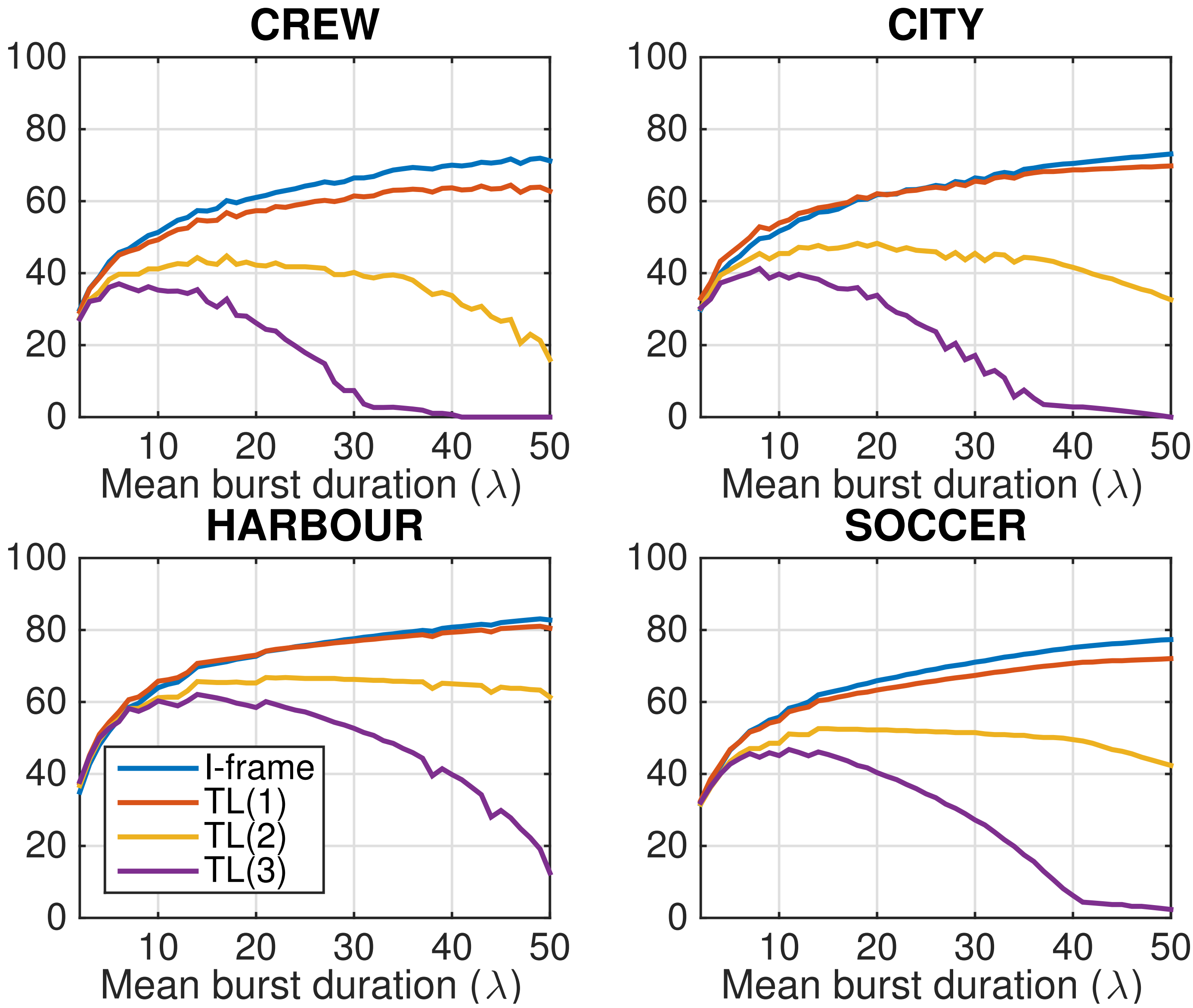

Fig. 15. Layer redundancy rates (%) for bursty packet losses at $R_S = 1600$ kbps (avg. FEC block size = 67 packets) and using hPP structure with 3 TLs.

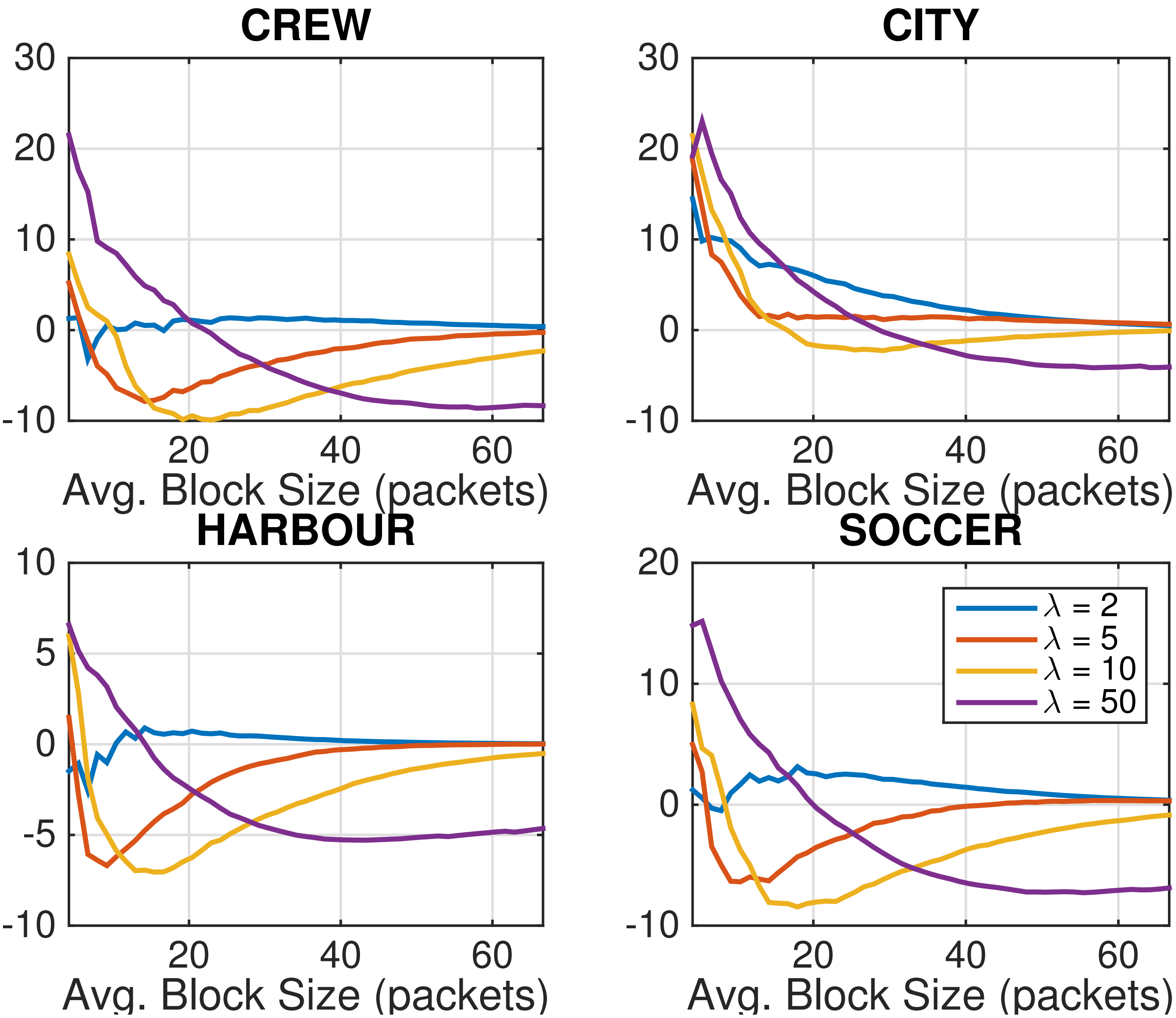

Fig. 16. Percentage of reduction in the achieved Q-STAR, relative to the IPP structure, given by $100 \times (1 - \overline{Q}_{\text{hPP}}/\overline{Q}_{\text{IPP}})$, in case of bursty losses with $\epsilon = 0.1$. Average frame size is calculated for each R_S, with $B = 200$ Bytes.

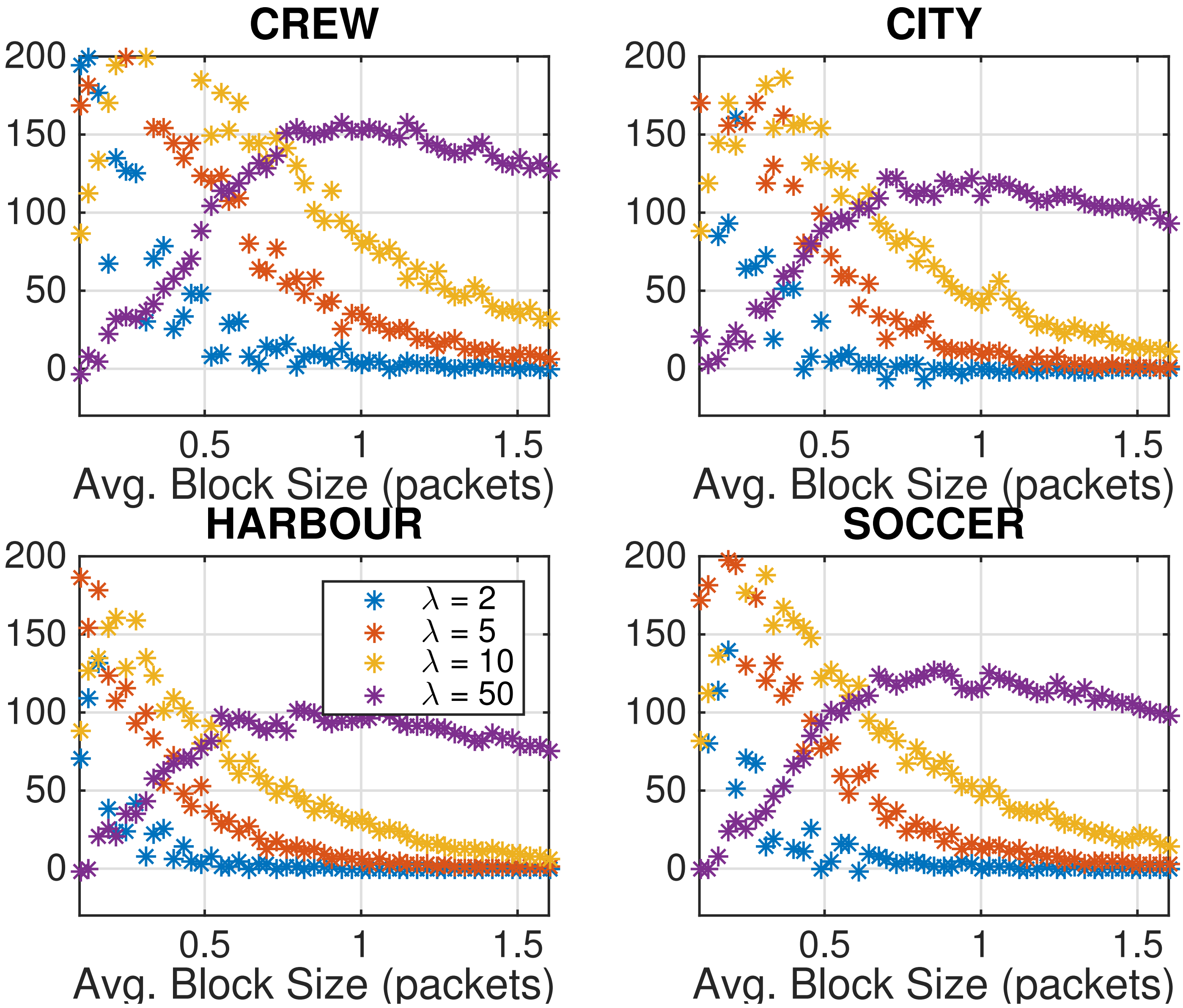

Fig. 17. Difference of mean frame intervals $\bar{\tau}_{\text{IPP}} - \bar{\tau}_{\text{hPP}}$ (msec) with IPP and hPP for bursty losses with $\epsilon = 0.1$, 10^5 simulations.

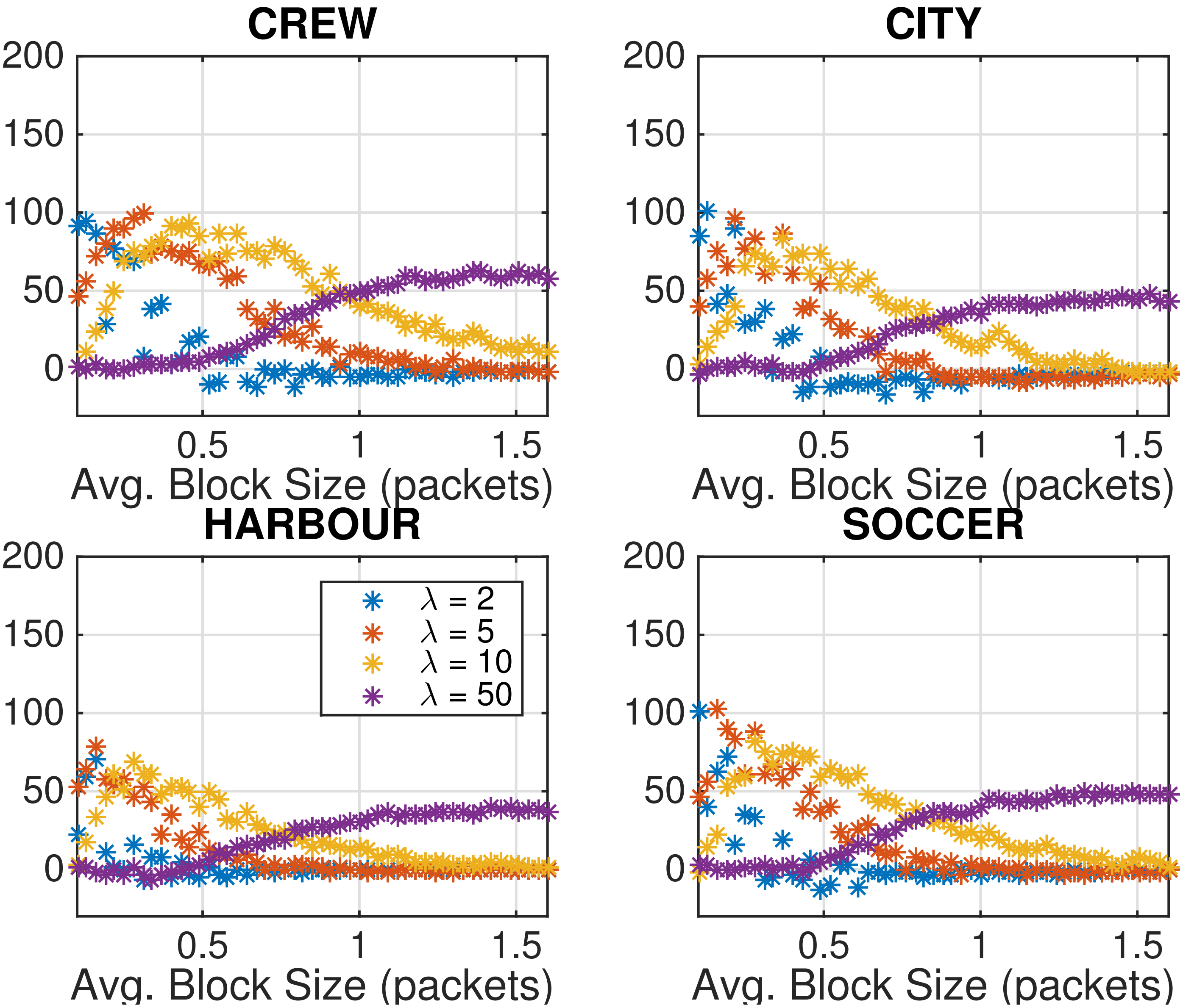

Fig. 18. Difference of standard deviations of frame intervals $\sigma_\tau^{\text{IPP}} - \sigma_\tau^{\text{hPP}}$ (msec) with IPP and hPP for bursty losses with $\epsilon = 0.1$, 10^5 simulations.