**Abstract**

- We consider the delay control problem for multiuser opportunistic scheduling policies.
- We develop a very effective, highly portable, and low complexity technique, the Opportunistic Bernoulli Mixing (OBM) scheme.
- OBM can support different user channel conditions and resource allocation requirements.
- OBM scheme can augment any temporally fair policy with a small set of algorithms.

**Delay Control via OBM**

For each time slot, perform the following user selection:
1. If at least one user reaches its deadline,
   - pick a random user \( k \) in the set of users reaching their deadlines,
   - serve user \( k \) with probability \( 1 - \nu_k \),
   - serve the opportunistic user with probability \( \nu_k \).
2. else if no user reaches its deadline, serve the opportunistic user.

**Transition Prob. Estimation**

For mathematical tractability, assume:
1. At any time slot, at most two users reach their deadlines.
2. Users reach their deadlines independently.

The transition probabilities can be estimated as:

\[
\alpha_k = \nu_k (1 - w_k) - 0.5 \sum_{j \neq k} \nu_j w_j - \frac{1}{2} |1 - \nu_k + \nu_k w_k| \tag{1}
\]

\[
\beta_k = (1 - w_k) + \sum_{j \neq k} (1 - \nu_j) \nu_j w_{j,d} \tag{2}
\]

\[
p_{j,d} = \gamma_j \beta_j^{T_{j,max}} \frac{1}{1 - \beta_j^{T_{j,max}} - 1} \tag{3}
\]

\[
\gamma_j = (1 - \alpha_j) p_{j,d} + (1 - \beta_j)(1 - p_{j,d}) \tag{4}
\]

where \( p_{j,d} \) is the deadline reaching probability for user \( j \).

**Finding OBM Parameters**

*Algorithm 1 Computing OBM Parameters*

1. Set \( \nu_k, \tau = 0 \), \( \nu_{k,u} = 1 \), \( \forall k \) and \( \nu = 0 \).
2. repeat
3. Compute \( q(\nu) = [q_1(\nu), \ldots, q_K(\nu)] \) using (5), (7), and (8).
4. for \( k = 1 \) to \( K \) do
5. if \( |q_k - \theta_k| > \epsilon \) and \( q_k < \theta_k \), set \( \nu_{k,1} = \nu_k \) and \( \nu_{k,2} = \frac{\nu_k + \theta_k}{2} \)
6. if \( |q_k - \theta_k| > \epsilon \) and \( q_k > \theta_k \), set \( \nu_{k,1} = \nu_k \) and \( \nu_{k,2} = \frac{\nu_k + \theta_k}{2} \)
7. else, keep \( \nu_k \) unchanged.
8. end for
9. until \( |q_k - \theta_k| \leq \epsilon \), \( \forall k \) or q(\nu) no longer changes

**Numerical Results**

- The fixed-point iteration method is used to find \( p_{j,d} \).
- \( p_k = f(p_k) \), where \( f_k(p_k) = \frac{\gamma_k(p_k) T_{k,max}}{1 - \alpha_k(p_k) \beta_k(p_k) T_{k,max} - 1} \).
- The PMF of service time \( \psi(\tau) \), with \( \tau = (m T_{k,max} + r) \), can be found in closed-form.
- \( \psi(\tau) = \begin{cases} (1 - p_{k,10} \psi(\tau) + \alpha_{k-1} \beta_{k-1} m) & \text{if } m > 0, \\ (1 - \beta_2 \beta_k) & \text{if } m = 0, \end{cases} \)
- Delay violation probability:
  \[ q_k(\nu) = \Pr[T_k > T_{k,max}] = 1 - \sum_{\tau=0}^{T_{k,max}} \psi(\tau) \]

Distribution of Service Delay for (K, Tmax, v) = (20, 30, 0.70)

Distribution of Service Delay for (K, Tmax, v) = (20, 30, 0.30)

Performance Comparisons for \( v = (0.1, 0.2, 0.3, 0.5, 0.7, 0.9) \)