Supplementary Materials

Appendix A
Proof of Theorem 1

We turn to prove the hardness of the corresponding decision problem of $(\mathbf{P1})$: given x, determine whether there exists a scheduling policy such that $\sum_{k=1}^{K} U_k(R_k) \geq x$. If we take $x = \sum_{k=1}^{K} U_k\left(\frac{1}{\mathsf{prd}_k}\right)$, since $R_k \leq \frac{1}{\mathsf{prd}_k}$, the decision problem is equivalent to the following feasibility-check problem: given timely throughput vector $\vec{R} = (R_1, \cdots, R_K) = (\frac{1}{\mathsf{prd}_1}, \cdots, \frac{1}{\mathsf{prd}_K})$, determine whether $\vec{R}$ is in the capacity region or not.

Note that our timely wireless flow problem can be regarded as a stochastic version of the preemptive scheduling problem of periodic real-time tasks on one processor [27]. In [27], the authors proved that the feasibility-check problem for arbitrary periodic real-time tasks on one processor is co-NP-hard in the strong sense. We largely follow their results to prove our Theorem 1.

In [27], authors leveraged the hardness result of the Simultaneous Congruences Problem (SCP): given a set $A = \{(a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)\} \subset \mathbb{N} \times \mathbb{N}^+$ and an integer $k \in [2, n]$, determine whether there are a subset $A' \subset A$ of k pairs and a positive integer x such that for every $(a_i, b_i) \in A'$, $x \equiv a_i \pmod{b_i}$. The hardness of SCP is shown in the following lemma.

***Lemma** 3 (Theorem 3.2, [27]):* SCP is NP-complete in the strong sense.

Now for any instance of SCP, i.e., $A = \{(a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)\} \subset \mathbb{N} \times \mathbb{N}^+$ and an integer $k \in [2, n]$, we construct (in polynomial time) the following instance of our feasibility-check problem: for any $i \in [1, K]$, let

$$(\mathsf{offset}_i, \mathsf{prd}_i, \mathsf{D}_i, \mathsf{B}_i) = ((k-1)a_i, (k-1)b_i, k-1, 1), \quad p_i = 1, R_i = \frac{1}{\mathsf{prd}_i} = \frac{1}{(k-1)b_i}. \tag{19}$$

Next we show that the timely throughput vector $\vec{R} = (R_1, \cdots, R_K) = (\frac{1}{(k-1)b_1}, \cdots, \frac{1}{(k-1)b_K})$ is in the capacity region if and only if the SCP answers No.

"If" Since the SCP answers No, then there are at most $k-1$ flows with a packet arrival at any slot. At slot 1, there are at most $k-1$ packets arriving in the system. All these packets can be scheduled with any work-conserving policy in the next $k-1$ slots during which no new packets arrive (since both $\mathsf{offset}_i = (k-1)a_i$ and $\mathsf{prd}_i = (k-1)b_i$ are a multiple of $k-1$ for any $i \in [1, K]$). After $k-1$ slots (i.e., at slot k), again there are at most $k-1$ packets arriving to the system, and thus the process can repeat. Therefore, all packets of all K flows can be delivered successfully before expiration. This yields to flow-i's timely throughput $\frac{1}{(k-1)b_i} = R_i$.

"Only if" Suppose that the SCP answers Yes, i.e, there exist a subset $A' \subset A$ of k pairs and a positive integer x such that $x \equiv a_i \pmod{b_i}, \forall (a_i, b_i) \in A'$, which equivalently means that $(k-1)x \equiv (k-1)a_i \pmod{(k-1)b_i}, \forall (a_i, b_i) \in A'$. Also, if we denote the common period $M = \text{Least.Common.Multiple}\{(k-1)b_i : (a_i, b_i) \in A'\}$, we easily obtain that $(k-1)x + jM \equiv (k-1)a_i \pmod{(k-1)b_i}, \forall (a_i, b_i) \in A'$ for any $j \in \mathbb{N}$. Thus, at slot $(k-1)x + jM + 1 (\forall j \in \mathbb{N})$, at least k packets arrive in the system, all of which had a deadline $k-1$. Thus, at least 1 packet will be expired every M slots, which means that at least a $\frac{1}{M} > 0$ timely throughput will be lost among all K flows. Therefore, $\vec{R} = (R_1, \cdots, R_K) = (\frac{1}{(k-1)b_1}, \cdots, \frac{1}{(k-1)b_K})$ cannot be achieved by any scheduling policy and thus is not in the capacity region.

Parts **"If"** and **"Only if"** prove that the timely throughput vector $\vec{R} = (R_1, \cdots, R_K) = (\frac{1}{(k-1)b_1}, \cdots, \frac{1}{(k-1)b_K})$ is in the capacity region if and only if the SCP answers No. Since SCP is NP-complete as shown in Lemma 3, we proves that it is co-NP-hard to determine whether or not the timely throughput vector $\vec{R} = (R_1, \cdots, R_K) = (\frac{1}{\mathsf{prd}_1}, \cdots, \frac{1}{\mathsf{prd}_K}) = (\frac{1}{(k-1)b_1}, \cdots, \frac{1}{(k-1)b_K})$ is in the capacity region. This completes the proof that the corresponding decision problem of $(\mathbf{P1})$ is co-NP-hard in the strong sense and thus $(\mathbf{P1})$ is also co-NP-hard in the strong sense.

Appendix B
Proof of Lemma 1

We notice that for any slot t, the following conditional independence relation among different flows holds,

$$P_t((\tilde{s}^1, \cdots, \tilde{s}^K)|(s^1, \cdots, s^K), a) = \prod_{k=1}^{K} P_t^k(\tilde{s}^k|s^k, a). \tag{20}$$

To show that

$$P_t((\tilde{s}^1, \cdots, \tilde{s}^K)|(s^1, \cdots, s^K), a) = P_{t'}((\tilde{s}^1, \cdots, \tilde{s}^K)|(s^1, \cdots, s^K), a), \tag{21}$$

where $t = l \cdot \mathsf{Prd} + \tau$ and $t' = (l+1) \cdot \mathsf{Prd} + \tau$, it suffices to show that

$$P_t^k(\tilde{s}^k|s^k, a) = P_{t'}^k(\tilde{s}^k|s^k, a), \forall k \in [1, K], s^k \in \mathcal{S}^k, a \in \mathcal{A}. \tag{22}$$

In the following, we will prove (22). Let us focus on an arbitrary $k \in [1, K]$. The proof will be divided into two steps. The first step is to show that the set of all possible states of flow k at slot t is *equal* to the set of all possible states of flow k at slot t', both of which are a subset of $\mathcal{S}^k$. The second step is to show that the transition probabilities at slots t and t' from any state $s = l_1^k l_2^k \cdots l_{\mathsf{D}_k}^k$ to any state $\tilde{s} = \tilde{l}_1^k \tilde{l}_2^k \cdots \tilde{l}_{\mathsf{D}_k}^k$ under the same action a are the same.

Step 1: (i) For any state in t, we can prove that it is also a state in t'. More specifically, if $l_i^k = 1$ under the state in t, we denote the corresponding packet as m. Clearly, if packet $m+1$ arrived the system and has not been delivered at t', the state at slot t' also has $l_i^k = 1$. Similar reasoning can be applied for $l_i^k = 0$. This proves that any state in t is also a possible state in t'. (ii) On the other hand, if $l_i^k = 1$ under the state in t', we denote the corresponding packet as m. Note that $t' \geq \tau + \mathsf{Prd}$, which means that[12] $m \geq 3$. Then, if packet $m - 1 \geq 2$ arrived the system and had not been delivered at t, the state at slot t also had $l_i^k = 1$. Similar reasoning can be

[12] See the definition of L in Lemma 1.

applied for $l_i^k = 0$. This proves that any state in t' is also a possible state in t. Thus, (i) and (ii) show that the sets of all possible state at slots t and t' are the same.

Step 2: The transition from state s to next state s' consists of three independent parts: (i) transmission, (ii) lead time evolution (including expiration), and (iii) arrival. Denote the lead-time-based binary string representation[13] after part (i) by $s_1 = m_1^k m_2^k \cdots m_{\mathsf{D}_k}^k$, after parts (i) and (ii) by $s_2 = n_1^k n_2^k \cdots n_{\mathsf{D}_k}^k$, and after parts (i), (ii) and (iii) by $s_3 = o_1^k o_2^k \cdots o_{\mathsf{D}_k}^k$. Part (i) characterizes the effect of transmission. If $l_i^k = 1$ and the corresponding packet is scheduled for transmission under action a, then $m_i^k = 0$ with probability p_k and $m_i^k = 1$ with probability $1-p_k$. If the corresponding packet for the bit l_i^k is not scheduled for transmission under action a, then $m_i^k = l_i^k$. Part (ii) characterizes the evolution of lead time. From the beginning of slot t to the beginning of slot $(t+1)$, the lead time of every packet will decrease by 1 and the packet with lead time 1 at slot t will expire at slot $(t+1)$. This evolution is equivalent to a left-shift operation for s_1. Namely, $s_2 = \mathsf{left\text{-}shift}(s_1)$, or $s_2 = n_1^k n_2^k \cdots n_{\mathsf{D}_k}^k = m_2^k m_3^k \cdots m_{\mathsf{D}_k}^k 0$. Part (iii) characterizes the effect of new arrival. At the beginning of slot $(t+1)$, it is possible that a new packet will come to the system if $(t+1)$ is the arrival time of some packet of flow k (see (1)). Note that the new arrived packet will always have lead time D_k and thus it only affects the last bit, i.e., $o_{\mathsf{D}_k}^k$. More specifically, if $(t+1)$ is not the arrival time of some packet, then $o_{\mathsf{D}_k}^k = 0$. If $(t+1)$ is the arrival time of some packet, then $o_{\mathsf{D}_k}^k = 1$ with probability B_k and $o_{\mathsf{D}_k}^k = 0$ with probability $1 - \mathsf{B}_k$. Other bits remain unchanged from part (ii) to part (iii), i.e., $s_3 = o_1^k o_2^k \cdots o_{\mathsf{D}_k-1}^k o_{\mathsf{D}_k}^k = n_1^k n_2^k \cdots n_{\mathsf{D}_k-1}^k o_{\mathsf{D}_k}^k$.

From the above analysis, we can easily see that parts (i) and (ii) are time-invariant and part (iii) is periodic with period Prd, which more specifically means that the arrival probability of a new packet at the beginning of slot $(t+1)$ is the same as that of at the beginning of slot $(t'+1 = t + \mathsf{Prd} + 1)$. Therefore, the transition probabilities at slots t and t' from any state $s = l_1^k l_2^k \cdots l_{\mathsf{D}_k}^k$ to any state $\tilde{s} = \tilde{l}_1^k \tilde{l}_2^k \cdots \tilde{l}_{\mathsf{D}_k}^k$ under the same action a are the same.

Step 1 and *Step 2* complete the proof for (22) and the whole proof of Lemma 1 is thus completed.

Appendix C
Proof of Theorem 2

For ease of exposition, we define the rate vector $\vec{R} \triangleq (R_1, R_2, \cdots, R_k)$ and the following two regions:

$$\mathcal{R}^{\text{RAC}} \triangleq \{\vec{R} : \vec{R} \text{ can be achieved by some RAC scheduling policy}\},$$

$$\mathcal{R}^{\text{LP}} \triangleq \{\vec{R} : \vec{R} \text{ (together with } \vec{x}\text{) is a feasible solution of (11a)} - \text{(11e)}\}.$$

Next we will prove the following two claims. **Claim 1:** $\mathcal{R}^{\text{RAC}} = \mathcal{R}^{\text{LP}}$. **Claim 2:** $\mathcal{R} = \mathcal{R}^{\text{LP}}$.

Clearly, **Claim 2** completes the proof for part (ii) of Theorem 2. And **Claim 1** & **Claim 2** show that $\mathcal{R} = \mathcal{R}^{\text{RAC}}$, which means that RAC can achieve any feasible rate vector. This completes the proof for part (i) of Theorem 2. Part (iii) of Theorem 2 will be proved when we prove **Claim 1**. The whole Theorem 2 is thus proved.

[13] Here we use different letters, i.e., m, n and o to represent the letter l for ease of exposition. They should be clear under the context here.

A. *Proof of **Claim 1:*** $\mathcal{R}^{RAC} = \mathcal{R}^{LP}$

Step 1: Show that $\mathcal{R}^{RAC} \subset \mathcal{R}^{LP}$. This is trivially true because of Definition 1 for an RAC scheduling policy. In Definition 1, for any RAC scheduling policy π, condition (i) specifies a conditional probability $\mathsf{Prob}_{A_t|S_t}(a|s)$, and condition (ii) specifies a series of state distribution $\mathsf{Prob}_{S_t}(s)$ where $\mathsf{Prob}_{S_{T_2+1}}(s) = \mathsf{Prob}_{S_{T_1+\mathsf{Prd}}}(s) = \mathsf{Prob}_{S_{T_1}}(s)$. Now if we define $x_t(s,a) = \mathsf{Prob}_{A_t|S_t}(a|s) \cdot \mathsf{Prob}_{S_t}(s), \forall t \in [T_1, T_2]$ and define $\vec{R}$ as the achieved rate vector of π, we can easily check that $\vec{R}$ together with $\vec{x}$ is a feasible solution of (11a) $-$ (11e). This proves that $\mathcal{R}^{\text{RAC}} \subset \mathcal{R}^{\text{LP}}$.

Step 2: Show that $\mathcal{R}^{LP} \subset \mathcal{R}^{RAC}$. We will base on the solution of the LP (11a)$-$(11e), denoted by $x_t(s,a)$ and $\vec{R}$, to construct an RAC scheduling policy such that the constructed RAC can achieve the rate vector $\vec{R}$. Note that the following construction is also the complete design of the RAC scheme in (12) and (13), and will also be used to prove part (iii) of Theorem 2. In the following, we define $\mathsf{Prob}_{A_t|S_t}(a|s) = \frac{x_t(s,a)}{\sum_{a' \in \mathcal{A}} x_t(s,a')}$ and $\mathsf{Prob}_{S_t}(s) = \sum_{a \in \mathcal{S}} x_t(s,a)$, same in (12) and (13), respectively. The construction contains two phases.

Phase 1: Initialization. The goal of this phase is to make the state distribution at slot T_1 to be the steady state distribution $\mathsf{Prob}_{S_{T_1}}(s)$ with the help of artificial *dummy packets*.

At the beginning of slot $T_0 \triangleq 1 + \max_{k \in [1,K]} \mathsf{offset}_k$, the arrival process of all flows has started. We first remove all arrived packets and then insert dummy packets randomly into the system such that the probability that the network state at slot T_0 is s is $\mathsf{Prob}_{S_{g(T_0)}}(s)$ for all $s \in \mathcal{S}$, where $g(T_0) = L \cdot \mathsf{Prd} + [1 + ((T_0 - 1) \mod \mathsf{Prd})]$ is the corresponding same-position slot in the optimized period $[T_1, T_2]$ in (11a) $-$ (11e). Note that to achieve such goal, we should set the lead time of the dummy packets carefully (and randomly), but any inserted flow-k dummy packet will expire at/before slot $T_0 + \mathsf{D}_k \le T_1$.

Phase 2: Making Scheduling Decisions. At the beginning of any slot $t \ge T_0$, we make the decision in the following way. Define $g(t) \triangleq L \cdot \mathsf{Prd} + [1 + ((t-1) \mod \mathsf{Prd})]$ as the corresponding same-position slot in the optimized period $[T_1, T_2]$ in (11a) $-$ (11e). Let s^t denote the current network state at time t, which counts both real packets and dummy packets. We use a *random scheduler* that chooses the flow k with probability

$$\mathsf{Prob}_{A_{g(t)}|S_{g(t)}}(k|s_t). \tag{23}$$

One can easily see that the resulting random process of the network state, i.e., $\{S_t : \forall t = 1, 2, \cdots\}$, is cyclostationary with period Prd, due to (11a) and (11b). Thus, the network state at slot $T_1 = L \cdot \mathsf{Prd} + 1$ also has the same distribution of the network state at slot 1, i.e., $\{\mathsf{Prob}_{S_{T_1}}(s) : s \in \mathcal{S}\}$. Note that here the network state $\{S_t\}$ counts both real packets and dummy packets. However, since any inserted flow-k dummy packet will expire at/before slot T_1 as shown in **Phase 1**, from slot T_1 on, only real packets exist in the system.

Now let us focus on the time interval $t \in [T_1, \infty)$. We can see that the scheduling strategy (23) satisfies condition (i) in Definition 1. And the random process of the network state, i.e., $\{S_{\tau_{\text{trans}}+t} : \forall t = 1, 2, \cdots\}$, is cyclostationary with period Prd. This shows that condition (ii) in Definition 1 also holds. Therefore, our two-phase construction indeed produces an RAC scheduling policy. Further, we can readily see that the constructed RAC can achieve the rate vector[14] $\vec{R}$ due to (11c), completing the proof for part (iii) of Theorem 2 and also the proof for $\mathcal{R}^{\text{LP}} \subset \mathcal{R}^{\text{RAC}}$.

Step 1 and *Step 2* complete the proof for **Claim 1**.

B. *Proof of **Claim 2**: $\mathcal{R} = \mathcal{R}^{LP}$*

Step 1: Show that $\mathcal{R}^{LP} \subset \mathcal{R}$. This is trivially true because $\mathcal{R}^{\text{LP}} = \mathcal{R}^{\text{RAC}} \subset \mathcal{R}$ from **Claim 1**.

Step 2: Show that $\mathcal{R} \subset \mathcal{R}^{LP}$. We prove this by showing that for any scheduling policy Ψ that achieves rate vector $\vec{R} = (R_1, \cdots, R_K) \geq 0$, we can always find a $\vec{x} = \{x_t(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}, t \in [T_1, T_2]\}$ such that $\vec{R}$ and $\vec{x}$ jointly satisfy (11a) – (11e).

For any scheduling policy Ψ, we choose an arbitrary integer $F \geq 1$ and compute the following quantity for all $t \in [1, \mathsf{Prd}]$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$,

$$y_t^F(s,a) \triangleq \frac{\sum_{f=0}^{F-1} \mathsf{E}\left\{1_{\{(\text{state,action}) = (s,a) \text{ at time } (t+f\cdot\mathsf{Prd})\}}\right\}}{F}, \quad (24)$$

where $1_{\{\cdot\}}$ is the indicator function. Intuitively, $y_t^F(s,a)$ in (24) quantifies how frequently we are going to see the network state being s and action being a at a (relative-position) slot t under policy Ψ, where the average is performed over both the probability space (due to the expectation operator) and over the time axis (due to the summation over different f's). Obviously, $y_t^F(s,a)$ depends on the underlaying policy Ψ. On the other hand, given Ψ and F, $y_t^F(s,a)$ always exists.

We observe that $y_t^F(s,a)$ in (24) is defined for all $t \in [1, \mathsf{Prd}]$. We can also define the $y_t^F(s,a)$ value for $t = \mathsf{Prd}+1$ in the following way,

$$y_{\mathsf{Prd}+1}^F(s,a) \triangleq \frac{\sum_{f=0}^{F-1} \mathsf{E}\left\{1_{\{(\text{state,action}) = (s,a) \text{ at time } (\mathsf{Prd}+1+f\cdot\mathsf{Prd})\}}\right\}}{F}. \quad (25)$$

It is worth emphasizing that in general $y_{\mathsf{Prd}+1}^F(s,a) \neq y_1^F(s,a)$ since the given policy Ψ may not be cyclostationary. We will use this statement soon.

Now we define the following

$$x_t^F(s,a) \triangleq y_{t-L\cdot\mathsf{Prd}}^F(s,a), \forall t \in [T_1, T_2], s \in \mathcal{S}, a \in \mathcal{A}. \quad (26)$$

Clearly, $x_t^F(s,a)$ is just a time shift of $y_t^F(s,a)$ such that the time indices are within the optimized period $[T_1, T_2]$ in (11a) – (11e).

We now find a sequence $\{F_n : n \geq 1\}$, which is a subsequence of the positive integer sequence $\{n : n \geq 1\}$, such that $\{y_t^{F_n}(s,a) : n \geq 1\}$ converges for *all* $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $t \in [1, \mathsf{Prd}]$. Toward that end, we first consider a tuple (s,a,t) where $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $t \in [1, \mathsf{Prd}]$. Since $y_t^n(s,a) \leq 1$, i.e., $\{y_t^n(s,a) : n \geq 1\}$ is a bounded sequence, we can find a convergent subsequence $\{y_t^{F_n^1}(s,a) : n \geq 1\}$. Clearly $\{F_n^1 : n \geq 1\}$ is a subsequence of $\{n : n \geq 1\}$. Now for another tuple (s',a',t') different from (s,a,t), since $\{y_{t'}^{F_n^1}(s',a') : n \geq 1\}$ is a bounded sequence, we can again find a convergent subsequence $\{y_{t'}^{F_n^2}(s',a') : n \geq 1\}$. Clearly $\{F_n^2 : n \geq 1\}$ is a subsequence of $\{F_n^1 : n \geq 1\}$, and both $\{y_t^{F_n^2}(s,a) : n \geq 1\}$ and $\{y_{t'}^{F_n^2}(s',a') : n \geq 1\}$ converge. By iterating over all $|\mathcal{S}|\cdot|\mathcal{A}|\cdot\mathsf{Prd}$ tuples for (s,a,t), we can find sequences $\{F_n^1 : n \geq 1\}$, $\{F_n^2 : n \geq 1\}$, $\cdots$, $\{F_n^{|\mathcal{S}|\cdot|\mathcal{A}|\cdot\mathsf{Prd}} : n \geq 1\}$, where $\{F_n^{i+1} : n \geq 1\}$ is a subsequence of $\{F_n^i : n \geq 1\}$ and $\{F_n^1 : n \geq 1\}$ is a subsequence of $\{n : n \geq 1\}$. We let $F_n = F_n^{|\mathcal{S}|\cdot|\mathcal{A}|\cdot\mathsf{Prd}}$, and thus $\{F_n : n \geq 1\}$ is a subsequence of $\{n : n \geq 1\}$. We also see that $\{y_t^{F_n}(s,a) : n \geq 1\}$ converges for *all* (s,a,t) tuples. We then define $y_t^\infty(s,a) \triangleq \lim_{n\to\infty} y_t^{F_n}(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $t \in [1, \mathsf{Prd}]$. We also define $x_t^\infty(s,a) \triangleq y_{t-\mathsf{Prd}}^\infty(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $t \in [T_1, T_2]$.

Now we will show that $x_t^\infty(s,a)$ and $\vec{R}$ can satisfy (11d), (11a), (11b) and (11c), respectively. Eq. (11e) is trivially true.

(i) It is easy to show that (11d) holds for $x_t^F(s,a)$ for any $F \geq 1$ (see (24)) and therefore (11d) holds for $x_t^\infty(s,a)$.

(ii) To show that (11a) holds, we should notice the following equation due to the total probability theorem where we define events $A(s') = \{\text{state} = s' \text{ at time } (t+1+f\cdot\mathsf{Prd})\}$ and $B(s,a) = \{(\text{state,action}) = (s,a) \text{ at time } (t+f\cdot\mathsf{Prd})\}$,

$$\begin{aligned}
&\mathsf{E}\{1_{\{A(s')\}}\} = \mathsf{Prob}(A(s')) \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \mathsf{Prob}(A(s')|B(s,a))\mathsf{Prob}(B(s,a)) \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_t(s'|s,a)\mathsf{Prob}(B(s,a)) \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_t(s'|s,a)\mathsf{E}\{1_{\{B(s,a)\}}\}.
\end{aligned} \quad (27)$$

Now due to the definition of $y_t^F(s,a)$ in (24), we get

$$\sum_{a\in\mathcal{A}} y_t^F(s,a) = \frac{\sum_{f=0}^{F-1} \mathsf{E}\left\{1_{\{\text{state} = s \text{ at time } (t+f\cdot\mathsf{Prd})\}}\right\}}{F}. \quad (28)$$

By inserting (27) into (28) (with slot and state modification) and doing some simple deductions, we get,

$$\sum_{a\in\mathcal{A}} y_{t+1}^F(s',a) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_t(s'|s,a)y_t^F(s,a). \quad (29)$$

This shows that (11a) holds for the time-shifted $y_t^F(s,a)$, i.e., $x_t^F(s,a)$, for any $F \geq 1$, and therefore (11a) holds for $x_t^\infty(s,a)$.

(iii) To show that (11b) holds, we cannot argue that it holds for $x_t^F(s,a)$ for any $F \geq 1$. That is because the following equation is generally not true,

$$y_1^F(s,a) = y_{\mathsf{Prd}+1}^F(s,a), \forall F \geq 1$$

Instead we will show that

$$y_1^\infty(s,a) = y_{\mathsf{Prd}+1}^\infty(s,a). \quad (30)$$

We first quantify the difference between $y_1^F(s,a)$ and $y_{\mathsf{Prd}+1}^F(s,a)$ with the following lemma.

[14] We can ignore the finite transient duration $[1, \tau_{\mathsf{trans}}] = [1, T_1 - 1]$ but only focus on the time interval $[T_1, \infty)$ to calculate the (long-term) timely throughput.

***Lemma** 4:* For any scheduling policy Ψ, we always have

$$|y_1^F(s,a) - y_{\mathsf{Prd}+1}^F(s,a)| \le \frac{1}{F}, \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

Proof: By the definitions of (24) and (25), we have

$$y_1^F(s,a) - y_{\mathsf{Prd}+1}^F(s,a) = \frac{\mathsf{E}\left\{1_{\{\text{(state,action) = }(s,a)\text{ at time }1\}}\right\}}{F} - \frac{\mathsf{E}\left\{1_{\{\text{(state,action) = }(s,a)\text{ at time }(1+F\cdot\mathsf{Prd})\}}\right\}}{F}. \tag{31}$$

Denote the right-hand side of (31) by $\mathsf{term}_1 - \mathsf{term}_2$. Since both term_1 and term_2 are non-negative, we have

$$|\mathsf{term}_1 - \mathsf{term}_2| \le \max\{\mathsf{term}_1, \mathsf{term}_2\}.$$

Since the numerators of term_1 and term_2 are upper bounded by 1, we have $\max\{\mathsf{term}_1, \mathsf{term}_2\} \le \frac{1}{F}$. The proof is thus completed. ∎

By Lemma 4, we thus conclude that (30) holds. Then from (30) and (29), we have

$$\sum_{a\in\mathcal{A}} y_1^\infty(s',a) = \sum_{a\in\mathcal{A}} y_{\mathsf{Prd}+1}^\infty(s',a) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_{\mathsf{Prd}}(s'|s,a) y_{\mathsf{Prd}}^\infty(s,a),$$

which shows that (11b) holds for $x_t^\infty(s,a)$.

(iv) Due to the definition of R_k in Sec. III-C, we have

$$\begin{aligned}
R_k &= \liminf_{\mathsf{T}\to\infty} \frac{\mathsf{E}\{\text{\# of flow-}k\text{ pkts delivered before exp. in }[1,\mathsf{T}]\}}{\mathsf{T}} \\
&= \liminf_{n\to\infty} \frac{\sum_{t=1}^{\mathsf{Prd}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r_k(s,a) y_t^n(s,a)}{\mathsf{Prd}} \\
&\le \lim_{n\to\infty} \frac{\sum_{t=1}^{\mathsf{Prd}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r_k(s,a) y_t^{F_n}(s,a)}{\mathsf{Prd}} \\
&= \frac{\sum_{t=1}^{\mathsf{Prd}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r_k(s,a) y_t^\infty(s,a)}{\mathsf{Prd}}.
\end{aligned} \tag{32}$$

Therefore, (11c) also holds for $x_t^\infty(s,a)$ and $\vec{R}$.

Thus, (i) – (iv) show that for any scheduling policy Ψ that achieves rate vector $\vec{R} = (R_1, \cdots, R_K)$, we can always find an $\vec{x} = \{x_t(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}, t \in [T_1, T_2]\}$ such that $\vec{R}$ and $\vec{x}$ are a feasible solution of (11a) – (11e).

This completes the proof for $\mathcal{R} \subset \mathcal{R}^{\mathsf{LP}}$.

Step 1 and *Step 2* complete the proof for **Claim 2**.

Appendix D
Proof of Lemma 2

We consider the random process $\{S_t : \forall t \ge 1\}$ for the system state induced by our RAC-Approx scheduling policy. We then consider the sub-process $\{Z_i : \forall i \ge 1\}$ where $Z_i \triangleq S_{i\cdot L\cdot\mathsf{prd}+1}$. Since the decision rules of our RAC-Approx scheduling policy using (17) and (18) repeat themselves every prd slots staring at slot $(L\cdot\mathsf{prd}+1)$, it is easy to see that $\{Z_i : \forall i \ge 1\}$ is a Markov chain.

We first show that the Markov chain $\{Z_i : \forall i \ge 1\}$ is an ergodic unichain [28]. We denote $\mathcal{S}_z$ as the state space of the Markov chain $\{Z_i : \forall i \ge 1\}$. Clearly $\mathcal{S}_z$ is a subset of the state space $\mathcal{S}$, and thus is finite. Then the Markov chain $\{Z_i : \forall i \ge 1\}$ must have at least one positive recurrent class $\mathcal{C}_z \subset \mathcal{S}_z$. We further denote the set of other states as $\bar{\mathcal{C}}_z = \mathcal{S}_z \backslash \mathcal{C}_z$. Since we assume that the packet arrival probability $\mathsf{B}_k < 1$, there exists a strictly positive probability that no packets of any flow arrive in the system during the time interval $[i\cdot L\cdot\mathsf{prd}+2, (i+1)\cdot L\cdot\mathsf{prd}+1]$. In addition, all packets arrived at slot $(i\cdot L\cdot\mathsf{prd}+1)$ will expire at slot $((i+1)\cdot L\cdot\mathsf{prd}+1)$ since $L\cdot\mathsf{Prd} \ge \max_{k\in[1,K]}(\mathsf{offset}_k + \mathsf{D}_k)$.[15] Therefore, it is possible from any state in $\mathcal{S}_z$ to the empty state, denoted as $s_\emptyset \in \mathcal{S}_z$, i.e.,

$$\mathsf{Prob}(Z_{i+1} = s_\emptyset | Z_i = s) > 0, \quad \forall s \in \mathcal{S}_z. \tag{33}$$

Now consider a positive recurrent state $s \in \mathcal{C}_z$. Since s is positive recurrent and $s_\emptyset$ is accessible from s due to (33), s must also be accessible from state $s_\emptyset$. Thus, $s_\emptyset$ and s communicate and we have that $s_\emptyset \in \mathcal{C}_z$. We further show that all states in $\bar{\mathcal{C}}_z$ are transient. Suppose that state $s' \in \bar{\mathcal{C}}_z$ is not transient, which means that it must be positive recurrent. Now again since $s_\emptyset$ is accessible from s' due to (33), s' must also be accessible from state $s_\emptyset$. Thus, $s_\emptyset$ and s' communicate and we have that $s' \in \mathcal{C}_z$, which is a contradiction that $s' \in \bar{\mathcal{C}}_z = \mathcal{S}_z \backslash \mathcal{C}_z$. Therefore, the Markov chain $\{Z_i : \forall i \ge 1\}$ has only one positive recurrent class $\mathcal{C}_z$, and thus is a unichain [28]. In addition, since $\mathsf{Prob}(Z_{i+1} = s_\emptyset | Z_i = s_\emptyset) > 0$, the empty state $s_\emptyset$ is aperiodic and thus the positive recurrent class $\mathcal{C}_z$ is also aperiodic. Therefore, the Markov chain $\{Z_i : \forall i \ge 1\}$ is an ergodic unichain.

Then according to Theorem 4.3.7 in [28], the Markov chain $\{Z_i : \forall i \ge 1\}$ has a unique steady state distribution, i.e., there exists a π_s such that

$$\lim_{i\to\infty} \mathsf{Prob}(Z_i = s) = \pi_s, \forall s \in \mathcal{S}_z. \tag{34}$$

Based on the initial state distribution $\{\pi_s : s \in \mathcal{C}_z\}$, for any flow k, we can get the expected number of delivered flow-k packets during the interval $[i\cdot L\cdot\mathsf{prd}+1, (i+1)\cdot L\cdot\mathsf{prd}+1]$ as $i \to \infty$, denoted by n_k. Then we can easily see that the achieved timely throughput for any flow k up to slot T, i.e., $\frac{\mathsf{E}\{\text{\# of flow-}k\text{ pkts delivered before expiration in }[1,\mathsf{T}]\}}{\mathsf{T}}$, converges to $\frac{n_k}{L\cdot\mathsf{prd}}$ as $\mathsf{T} \to \infty$. This completes the proof.

Appendix E
How to Solve (P4) and (P5) Approximately in Polynomial Time by Unwinding Traffic Patterns

For any flow k with A&E profile $(\mathsf{offset}_k, \mathsf{prd}_k, \mathsf{D}_k, \mathsf{B}_k)$ and $\mathsf{prd}_k < \mathsf{D}_k$, we unwind it into $\lceil \mathsf{D}_k/\mathsf{prd}_k \rceil$ flows, indexed from k_1, k_2, to $k_{\lceil \mathsf{D}_k/\mathsf{prd}_k \rceil}$: flow k_i $(1 \le i \le \lceil \mathsf{D}_k/\mathsf{prd}_k \rceil)$ has A&E profile $(\mathsf{offset}_{k_i}, \mathsf{prd}_{k_i}, \mathsf{D}_{k_i}, \mathsf{B}_{k_i})$ where

$$\begin{aligned}
\mathsf{offset}_{k_i} &= \mathsf{offset}_k + (i-1)\mathsf{prd}_k, & \mathsf{D}_{k_i} &= \mathsf{D}_k, \\
\mathsf{prd}_{k_i} &= \lceil \mathsf{D}_k/\mathsf{prd}_k \rceil \mathsf{prd}_k, & \mathsf{B}_{k_i} &= \mathsf{B}_k.
\end{aligned}$$

One unwinding example is shown in Fig. 9. Now for each flow k_i, the delay D_{k_i} is no more than the period prd_{k_i}. Thus, the size of the state space of flow k_i can be reduced to 2 by using the compression method in (10). In addition, any flow-k_i's packet will be successfully delivered with probability p_k.

[15] See the definition of L in Lemma 1.

After such unwinding, in the new system, all flows have size-2 state space. We further denote R_{k_i} as the timely throughput of flow k_i in the new system, and define

$$R_k = \sum_{i=1}^{\lceil \mathsf{D}_k/\mathsf{prd}_k \rceil} R_{k_i},$$

as the timely throughput for flow k in the original system. Now if we use the relaxation method in Sec. VI-A for the new system, the complexity (in terms of number of variables and constraints) of solving ($\mathbf{P4}$) and ($\mathbf{P5}$) becomes $O((\sum_{k=1}^{K}\lceil \mathsf{D}_k/\mathsf{prd}_k \rceil) \cdot K \cdot \mathsf{Prd})$. Therefore, by unwinding traffic patterns, we can solve ($\mathbf{P4}$) and ($\mathbf{P5}$) for the original system *approximately* in polynomial time.

Appendix F
The Optimal RAC Policy for the Setting in Fig. 5 in Sec. VII-B

In this part, we show the optimal RAC policy for the setting in Fig. 5 in Sec. VII-B, i.e.,

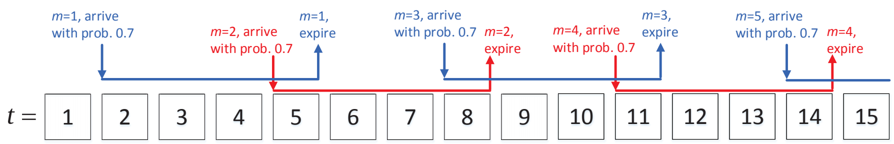
$$(\mathsf{offset}_1, \mathsf{prd}_1, \mathsf{D}_1, \mathsf{B}_1, p_1) = (0,4,4,1,0.5), U_1(R_1) = \log(R_1),$$
$$(\mathsf{offset}_2, \mathsf{prd}_2, \mathsf{D}_2, \mathsf{B}_2, p_2) = (2,4,4,1,0.5), U_2(R_2) = \log(R_2),$$
$$(\mathsf{offset}_3, \mathsf{prd}_3, \mathsf{D}_3, \mathsf{B}_3, p_3) = (0,1,3,0.9,0.7), U_3(R_3) = \log(R_3).$$

In Tab. IV, we show the conditional probability $\mathsf{Prob}_{A_t|S_t}(a|s)$ for each state $s = (s^1, s^2, s^3)$, each action $a \in \{1,2,3\}$, and each slot $t \in [T_1, T_2] = [9, 12]$, which is the optimization period in ($\mathbf{P2}$).

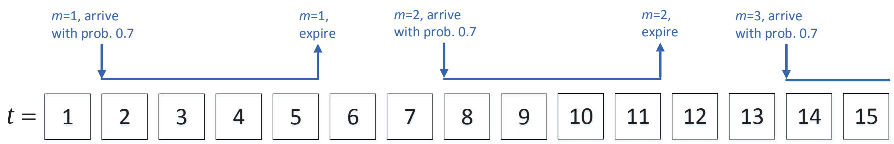
Note that since $\mathsf{D}_k \le \mathsf{prd}_k$ for $k = 1, 2$, we can use the compression method similar in (10) to denote the state of flow 1 and flow 2. More specifically, we use $s^1 = 1$ (resp. 0) to denote the flow 1's state: flow 1 has one (resp. no) packet, and we use $s^2 = 1$ (resp. 0) to denote the flow 2's state: flow 2 has one (resp. no) packet. We remark that though we do not have the lead time information in the state representation for flow 1 and flow 2, we can recover the lead time from the current slot. For flow 3, since $\mathsf{D}_k > \mathsf{prd}_k$, we still use the lead-time-representation to denote its state, i.e., $s^3 = l_1^3 l_2^3 l_3^3$ where $l_i^k = 1$ (resp. 0) is the state that flow 3 has one (resp. no) packet with lead time i. For example, state $s = (s^1, s^2, s^3) = (0, 1, 001)$ means that in the AP's queue, there is no flow-1 packet, one flow-2 packet, and one flow-3 packet with lead time 3.

Now we show several examples to illustrate the optimal RAC policy in Tab. IV.

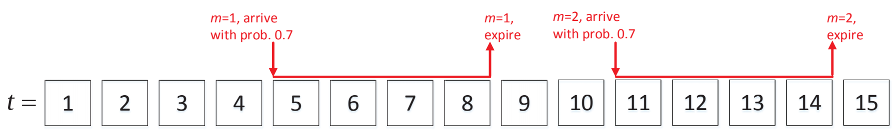
- Consider state $s = (s^1, s^2, s^3) = (0, 0, 000)$ at slot $t = 9$, i.e., all flows have no packet. This state is impossible (we use the hyphen (-) notation in Tab. IV to denote an impossible state at the corresponding slot) because at the beginning of slot 9, flow 1 has a packet arrival with probability $\mathsf{B}_1 = 1$ and thus the flow-1 state must be 1.
- Consider state $s = (s^1, s^2, s^3) = (0, 0, 000)$ at slot $t = 10$, i.e., all flows have no packet. Though our optimal RAC policy will still schedule each flow with equal probability (0.33), no packets will be transmitted and thus the system actually remains idle.
- Consider state $s = (s^1, s^2, s^3) = (0, 1, 100)$ at slot $t = 11$, i.e., flow 2 has a packet with lead time 4 and flow 3 has a packet with lead time 1. Our optimal RAC policy will schedule flow 3 with probability 1, i.e, give the complete priority to flow 3. This is reasonable because the flow-3 packet will be expired in the next slot and thus is more urgent than the flow-2 packet.
- Consider state $s = (s^1, s^2, s^3) = (1, 0, 111)$ at slot $t = 12$, i.e., flow 1 has a packet with lead time 1, and flow 2 has a packet with lead time 1, a packet with lead time 1, and a packet with lead time 3. Our optimal RAC policy will schedule flow 1 with probability 0.52 and schedule flow 3 with probability 0.48, i.e., gives a little bit more priority to flow 1. This is reasonable because though both flows have a very urgent packet (with lead time 1), flow 3 has more packet candidates than flow 1.



(a) Original flow k with A&E profile: $(\mathsf{offset}_k, \mathsf{prd}_k, \mathsf{D}_k, \mathsf{B}_k) = (1, 3, 4, 0.7)$



(b) Unwound flow k_1 with A&E profile: $(\mathsf{offset}_{k_1}, \mathsf{prd}_{k_1}, \mathsf{D}_{k_1}, \mathsf{B}_{k_1}) = (1, 6, 4, 0.7)$



(c) Unwound flow k_2 with A&E profile: $(\mathsf{offset}_{k_2}, \mathsf{prd}_{k_2}, \mathsf{D}_{k_2}, \mathsf{B}_{k_2}) = (4, 6, 4, 0.7)$

Fig. 9. An illustrating example of unwinding traffic patterns. We unwind the original flow k in (a) into flow k_1 in (b) and flow k_2 in (c).

TABLE IV

THE OPTIMAL RAC POLICY FOR THE SETTING IN FIG. 5 IN SEC. VII-B, I.E., $(\mathsf{offset}_1, \mathsf{prd}_1, \mathsf{D}_1, \mathsf{B}_1, p_1) = (0, 4, 4, 1, 0.5), U_1(R_1) = \log(R_1)$, $(\mathsf{offset}_2, \mathsf{prd}_2, \mathsf{D}_2, \mathsf{B}_2, p_2) = (2, 4, 4, 1, 0.5), U_2(R_2) = \log(R_2)$, AND $(\mathsf{offset}_3, \mathsf{prd}_3, \mathsf{D}_3, \mathsf{B}_3, p_3) = (0, 1, 3, 0.9, 0.7), U_3(R_3) = \log(R_3)$. WE SHOW THE CONDITIONAL PROBABILITY $\mathsf{Prob}_{A_t|S_t}(a|s)$ FOR EACH STATE $s = (s^1, s^2, s^3)$, EACH ACTION $a \in \{1, 2, 3\}$ IN THE PERIOD $[T_1, T_2] = [9, 12]$. **NOTE 1:** THE FLOW-1 STATE $s^1 = 1$ (RESP. 0) MEANS ONE (RESP. NO) FLOW-1 PACKET; THE FLOW-2 STATE $s^2 = 1$ (RESP. 0) MEANS ONE (RESP. NO) FLOW-2 PACKET; THE FLOW-3 STATE $s^3 = l_1^3 l_2^3 l_3^3$ WHERE $l_i^k = 1$ (RESP. 0) MEANS ONE (RESP. NO) FLOW-3 PACKET WITH LEAD TIME i. FOR EXAMPLE, STATE $s = (s^1, s^2, s^3) = (0, 1, 001)$ MEANS THAT IN THE AP'S QUEUE, THERE IS NO FLOW-1 PACKET, ONE FLOW-2 PACKET, AND ONE FLOW-3 PACKET WITH LEAD TIME 3. **NOTE 2:** THE HYPHEN (-) NOTATION MEANS THAT THE CORRESPONDING STATE IS IMPOSSIBLE AT THE CORRESPONDING SLOT UNDER THIS OPTIMAL RAC POLICY. **NOTE 3:** WE ONLY SHOW 2 DIGITS AFTER THE DECIMAL POINT. THUS $\sum_{a=1}^{3} \mathsf{Prob}_{A_t|S_t}(a|s)$ MAY NOT BE 1.

	$t = T_1 = 9$			$t = 10$			$t = 11$			$t = T_2 = 12$		
$s = (s^1, s^2, s^3)$	$a = 1$	$a = 2$	$a = 3$	$a = 1$	$a = 2$	$a = 3$	$a = 1$	$a = 2$	$a = 3$	$a = 1$	$a = 2$	$a = 3$
(0, 0, 000)	-	-	-	0.33	0.33	0.33	-	-	-	0.33	0.33	0.33
(0, 0, 001)	-	-	-	0.00	0.00	1.00	-	-	-	0.00	0.00	1.00
(0, 0, 010)	-	-	-	0.00	0.00	1.00	-	-	-	0.00	0.00	1.00
(0, 0, 011)	-	-	-	0.00	0.00	1.00	-	-	-	0.00	0.00	1.00
(0, 0, 100)	-	-	-	-	-	-	-	-	-	-	-	-
(0, 0, 101)	-	-	-	0.00	0.00	1.00	-	-	-	0.00	0.00	1.00
(0, 0, 110)	-	-	-	0.00	0.00	1.00	-	-	-	0.00	0.00	1.00
(0, 0, 111)	-	-	-	0.00	0.00	1.00	-	-	-	0.00	0.00	1.00
(0, 1, 000)	-	-	-	-	-	-	0.00	1.00	0.00	0.00	1.00	0.00
(0, 1, 001)	-	-	-	0.00	0.98	0.02	0.00	0.83	0.17	0.00	0.85	0.15
(0, 1, 010)	-	-	-	0.00	1.00	0.00	0.00	0.03	0.97	0.00	0.01	0.99
(0, 1, 011)	-	-	-	0.00	0.97	0.03	0.00	0.48	0.52	0.00	0.62	0.38
(0, 1, 100)	-	-	-	-	-	-	0.00	0.00	1.00	0.00	0.00	1.00
(0, 1, 101)	-	-	-	0.00	0.50	0.50	0.00	0.01	0.99	0.00	0.00	1.00
(0, 1, 110)	-	-	-	0.00	0.50	0.50	0.00	0.00	1.00	0.00	0.00	1.00
(0, 1, 111)	-	-	-	0.00	0.52	0.48	0.00	0.00	1.00	0.00	0.87	0.13
(1, 0, 000)	1.00	0.00	0.00	1.00	0.00	0.00	-	-	-	-	-	-
(1, 0, 001)	0.83	0.00	0.17	0.85	0.00	0.15	-	-	-	0.98	0.00	0.02
(1, 0, 010)	0.03	0.00	0.97	0.01	0.00	0.99	-	-	-	1.00	0.00	0.00
(1, 0, 011)	0.48	0.00	0.52	0.62	0.00	0.38	-	-	-	0.97	0.00	0.03
(1, 0, 100)	0.00	0.00	1.00	0.00	0.00	1.00	-	-	-	-	-	-
(1, 0, 101)	0.01	0.00	0.99	0.00	0.00	1.00	-	-	-	0.50	0.00	0.50
(1, 0, 110)	0.00	0.00	1.00	0.00	0.00	1.00	-	-	-	0.50	0.00	0.50
(1, 0, 111)	0.00	0.00	1.00	0.87	0.00	0.13	-	-	-	0.52	0.00	0.48
(1, 1, 000)	0.00	1.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
(1, 1, 001)	0.01	0.99	0.00	0.00	1.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
(1, 1, 010)	0.00	0.37	0.63	0.00	1.00	0.00	0.37	0.00	0.63	1.00	0.00	0.00
(1, 1, 011)	0.00	0.50	0.50	0.22	0.39	0.38	0.50	0.00	0.50	0.39	0.22	0.38
(1, 1, 100)	0.00	0.00	1.00	0.00	0.52	0.48	0.00	0.00	1.00	0.52	0.00	0.48
(1, 1, 101)	0.00	0.00	1.00	0.00	0.65	0.35	0.00	0.00	1.00	0.65	0.00	0.35
(1, 1, 110)	0.00	0.55	0.45	0.00	0.62	0.38	0.55	0.00	0.45	0.62	0.00	0.38
(1, 1, 111)	0.00	0.91	0.09	0.14	0.77	0.09	0.91	0.00	0.09	0.77	0.14	0.09