Semi-Persistent Scheduling for 5G Downlink Based on Short-Term Traffic Prediction

Qing He*, György Dán*, Georgios P. Koudouridis†
*Division of Network and Systems Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
email: qhe@kth.se, gyuri@kth.se
†RAN System Lab., Stockholm Research Center, Huawei Technologies, Stockholm, Sweden
email: george.koudouridis@huawei.com

***Abstract*—Efficient communication and computing resource allocation is becoming a fundamental issue in wireless networks. Efficiency is most often defined in terms of throughput, utilization and spectral efficiency, while the required computational effort is often overlooked. In this paper, we focus on efficient and computationally lightweight downlink scheduling, and we propose a semi-persistent scheduler based on adaptive short term traffic prediction. We evaluate the performance of the proposed scheduler in terms of throughput, fairness, latency, and scheduling complexity. Our numerical results show that scheduling with prediction is a promising approach in improving network performance. The proposed semi-persistent scheduler performs equally well in terms of throughput, fairness, and latency as traditional proportional-fair scheduling, but at a significantly reduced computational cost.**

***Index Terms*—Scheduling, traffic prediction, wireless networks**

I. Introduction

Efficient use of spectrum and computational resources is becoming increasingly important in the design of future mobile networks, so as to cope with increasing traffic demands despite limited spectrum availability and computational capability. A fundamental solution for achieving efficiency is scheduling, which in general refers to processes that determine the amount of wireless resources that are allocated to each user. The resulting schedule typically depends on many factors including channel conditions, interference, data rates, delay, and buffer status, and is chosen so as to optimize performance criteria, such as throughput, fairness, energy consumption, or their combination.

Cellular wireless networks employ a variety of scheduling algorithms, including round-robin (RR), proportional fair (PF) [1], and semi-persistent scheduling (SPS) [2]. RR is a simple scheduling algorithm following the time division multiple access (TDMA) scheme, and schedules users one-by-one in a cyclic order. RR gives equal opportunity to the users to transmit and thus it ensures fairness, but it may result in poor overall throughput. PF is a widely used dynamic scheduler that balances between total network throughput and fairness. A PF scheduler usually operates on a per transmission time interval (TTI) basis. Such a dynamic scheduler can adapt to fast changing channel conditions, but it comes with high computational cost and control signaling overhead [2]. SPS was initially designed for uplink scheduling for voice over Internet Protocol (VoIP) [3]. The principle of SPS is to allocate radio resources to a user for multiple consecutive TTIs, and thus reducing the computational cost and control signaling overhead. In comparison to dynamic scheduling, SPS is more computationally efficient. Nonetheless, the major drawback of SPS is that it may not be able to adjust the scheduling decision promptly enough, to accommodate time-varying traffic demands.

Although SPS was originally proposed for improving uplink scheduling performance for VoIP, we argue that it could effectively be used for reducing the computational cost and the control channel overhead for downlink scheduling as well. In this paper we propose a novel predictive semi-persistent scheduler that is computationally efficient and at the same time achieves high throughput, fairness and low latency. The proposed semi-persistent downlink scheduler pre-allocates radio resources to mobile users based on a short-term prediction of arriving traffic over multiple TTIs. We propose two priority functions that take into consideration the instantaneous channel conditions, historical data rates, the buffer occupancy and the predicted traffic to assign a scheduling priority to each user. We evaluate the proposed predictive SPS algorithms by simulations, and our results show that the SPS scheduler achieves equally good performance as a dynamic PF scheduler, but at a significantly reduced computational cost.

The rest of the paper is organized as follows. In Section II we review the related work for wireless network scheduling. In Section III we define the scheduling problem and propose the SPS algorithm. In Section IV we design the priority functions. In Section V we introduce the traffic predictor and the proposed architecture of the scheduler, following with performance evaluation in Section VI. Finally, we summarize the findings and discuss potential extensions in Section VII.

II. Related Work

The scheduling problem in wireless systems has been studied extensively in the past (see for example [1], [4], [5] for a detailed exposition). The early works starting in the 1980s adopted network-level approaches with various medium access control (MAC) protocols that are proposed to achieve better throughput [6], [7]. Under the so called protocol model, the interference among links is described by a conflict graph, where a transmission is successful if no adjacent node of the receiver is active. To solve the scheduling problem, graph-based algorithms employing implicit or explicit coloring strategies were widely used (e.g., [8], [9]). Later on, the scheduling problem was studied under the interference model, which couples the MAC with the physical layer and leads to a cross-layer view of transmission rate and access control [10]. This approach incorporates the effect of power control and channel conditions on the achievable transmission rates. Furthermore, existing results indicate that it is possible to integrate the problems of routing, scheduling, and physical layer effects leading to structural results such as the back-pressure algorithm [11]. The scheduling problem under the interference model has been proved to be NP-hard with arbitrary gain and with geometry-based gain, in [12] and [13], respectively. A variety of scheduling protocols and algorithms have been proposed to various wireless systems.

SPS was originally designed to reduce the control channel overhead for VoIP based services in the uplink of LTE networks [3]. The core idea of SPS is to significantly reduce heavy load on physical downlink control channel (PDCCH) by doing minimum downlink assignment and uplink grant. There are different approaches to SPS and algorithmic variations in the related work. For instance, in [14] a predictive SPS scheme that takes advantage of transmission history is proposed to effectively reduce the uplink latency of LTE systems. In [15], the authors designed an adaptive SPS algorithm for Machine-to-Machine (M2M) communications by utilizing the device buffer information. Since SPS allocates radio resources for a longer period of time, a good prediction on the incoming traffic is of critical importance. In addition, SPS can be improved if channel prediction is available [16], [17]. We remark that the channel prediction itself is a challenging task and here we focus on how to improve the efficiency of scheduling by traffic prediction. Compared to existing works that focus mostly on uplink scheduling and channel prediction, in this paper we investigate the potential of SPS for the cellular downlink, by proposing traffic prediction-aware extensions of the widely-used PF scheduler.

III. Semi-Persistent Downlink Scheduling

We consider a cellular downlink serving N user equipments (UE) over a number C of subcarriers. Let us denote by $\mathcal{C}$ the set of the subcarriers and $\mathcal{N}$ the set of users. The traditional scheduling problem is to decide in each TTI the subset of UEs that should transmit, and the subcarrier they should use for transmission. The selection of UEs could be done based on a user priority function P, which is used for computing the scheduling priority of each UE. Unlike for dynamic scheduling, where the user priority P_n for user n is computed in each TTI, in SPS a scheduler can allocate radio resources for a sequence of TTIs. Let us denote by m the number of TTIs over which a scheduling decision will be made persistent. We denote by $\mathcal{C}_S$ the set of subcarriers for which SPS is used, and by $C' = |\mathcal{C}_S| \leq C$ the number of such subcarriers. Without loss of generality, we index by $\{1, 2, \ldots, C'\}$ the subcarriers using SPS. The subcarriers $C' + 1, \ldots, C$ may employ dynamic scheduling in order to maintain a good service quality for users in fast changing environments. We define $\mathcal{C}_D = \mathcal{C} \setminus \mathcal{C}_S$ to be the set of such subcarriers.

Algorithm 1 Prediction-based Downlink SPS

Input: $\mathcal{N}$, $\mathcal{C}$, $\mathcal{C}_S$, $\mathcal{C}_D$, m, P, $B_n(t_0)$
Output: $U_S(t)$, $U_D(t)$

1: $t \leftarrow t_0$, $\mathcal{C}_S^a \leftarrow \mathcal{C}_S$, $U_S'(t) \leftarrow \emptyset$, $U_S''(t) \leftarrow \emptyset$, $U_D(t) \leftarrow \emptyset$
2: $U \leftarrow \{n \in \mathcal{N} : B_n(t) \neq 0\} \setminus U_S'(t)$
3: Compute P_n, $\forall n \in U$, and sort U in the descending order of P_n
4: $U_S''(t)_i \leftarrow U_i$, $i = 1, 2, \ldots, |\mathcal{C}_S^a|$
5: $U_D(t)_j \leftarrow U_{j+|\mathcal{C}_S^a|}$, $j = 1, 2, \ldots, |\mathcal{C}_D|$
6: $U_S(t) \leftarrow U_S'(t) \cup U_S''(t)$, $m(n) \leftarrow m$, $\forall n \in U_S''(t)$
7: $U_S'(t+1) \leftarrow U_S(t)$, $U_S''(t+1) \leftarrow \emptyset$
8: $t \leftarrow t+1$, $m(n) \leftarrow m(n) - 1$, $\forall n \in U_S'(t)$
9: Update $B_n(t)$, $\forall n \in \mathcal{N}$
10: **for** $n \in U_S'(t)$ **do**
11:     **if** $m(n) = 0$ or $B_n(t) = 0$ **then**
12:         $U_S'(t) \leftarrow U_S'(t) \setminus \{n\}$; update $\mathcal{C}_S^a$
13: repeat lines 2-12 until $t = T$
14: **return** $U_S(t)$, $U_D(t)$

The proposed SPS algorithm then works as follows. Upon every TTI the algorithm updates the set $U \subseteq \{1, 2, \ldots, N\}$ of users requiring radio resources. The set U contains users that have data to transmit and are not currently assigned to any of the subcarriers $\mathcal{C}_S$. It then updates the set of subcarriers $\mathcal{C}^a = \mathcal{C}_S^a \cup \mathcal{C}_D$ that can be assigned, where $\mathcal{C}_S^a \subseteq \mathcal{C}_S$ is the set of subcarriers for which SPS is used but are not assigned. The algorithm then calculates the user priority for each user $u \in U$, using one of the user priority functions presented in Section IV. Based on the scheduling priority of each user, the SPS algorithm allocates the $|\mathcal{C}_S^a|$ highest ranked users to subcarriers $\mathcal{C}_S^a$ for m TTIs, and the next $|\mathcal{C}_D|$ highest ranked users to subcarriers $\mathcal{C}_D$ for a single TTI. A user u that was scheduled semi-persistently to a subcarrier $c \in \mathcal{C}_S$ but has no data to transmit at the end of a TTI is removed from the subcarrier, and c is added to $\mathcal{C}_S^a$.

In Algorithm 1 we show the pseudo-code of the proposed scheduling algorithm. In the pseudo-code we denote by $U_S(t)$ the set of users being persistently scheduled in the t^{th} TTI, by $U_D(t)$ the users dynamically scheduled in the t^{th} TTI, and by $B_n(t)$ the backlogged data of UE n.

IV. Predictive User Priority Functions

We now propose two predictive user priority functions for SPS. We start the exposition with the widely used PF scheduler for cellular downlink scheduling, which maintains a balance between two competing objectives: maximizing total throughput of the network and throughput fairness.

A. Proportional Fair Priority Function

The user priority computed by the PF scheduler is inversely proportional to the anticipated resource consumption of the user [18], and is computed for UEs for which there is data to be transmitted as

$$P_n = \frac{T_n^{\alpha}}{R_n^{\beta}}, \tag{1}$$

where T_n denotes the estimated achievable data rate of UE n in the present time slot, and R_n is the past average data rate for UE n computed over a so called latency window, whose length t_c is related to the maximum time that a user can be starved. The parameters $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ can be used to trade-off between throughput and fairness. In particular, a high value of α would result in serving the UE with the best channel conditions more often and maximize throughput, while a high value of β would lead to serving the UE with poor channel condition often enough so that they would have an acceptable average rate.

Clearly, PF scheduling does not take into consideration future traffic arrivals and user buffer occupancy. In fact, the scheduling algorithm makes the assumption that there is always sufficient backlogged data in the user buffer and the buffer size is unlimited. As currently video accounts for more than 80% of global internet traffic [19], the downlink traffic is likely to be very bursty. Burstiness, in turn could result in significant delays when the user priority (1) is low. Motivated by the above, in what follows we propose two novel priority functions that take into account the predicted traffic arrival and buffer size.

B. Backlog-stable Predictive Priority Function

The first priority function introduces a multiplicative term that captures the ratio of the backlog at the end of the prediction interval and the average past backlog of the user,

$$P_n^B = \frac{T_n^{\alpha}}{R_n^{\beta}} \left(\frac{A_n + B_n}{\bar{B}_n} \right)^{\gamma}, \tag{2}$$

where A_n is the predicted amount of data that will arrive in the next m TTIs to UE n, and B_n is the amount of buffered data for UE n in the current TTI. Thus, the numerator $A_n + B_n$ corresponds to the expected backlog after m TTIs if this user is not scheduled. $\bar{B}_n$ is the average past buffer size for the UE computed based on a sliding window, similar to R_n. Finally, γ is a tuning parameter. We refer to the SPS algorithm using the priority function defined in (2) as SPS-I.

The scheduler can be tuned by adjusting the values of α, β, and $0 \leq \gamma \leq 1$. Clearly, for $\gamma = 0$ the priority function is the same as (1). On the contrary, for $\alpha = \beta = 0$ the scheduler ignores the instantaneous rate and the channel conditions, and the UE's scheduling priority will solely depend on the incoming traffic and buffer size, akin to longest-queue-first scheduling.

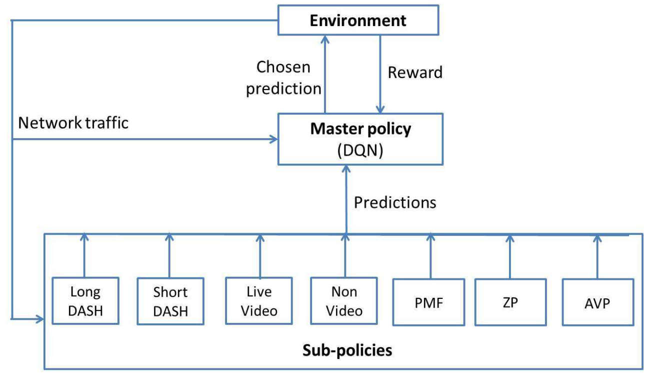


Figure 1. The meta-learning scheme used for short-term traffic prediction. The master policy uses reinforcement learning for choosing the sub-policy that is expected to predict most accurately for the next time interval [20].

C. Rate-stable Predictive Priority Function

The second priority function is inspired by the idea of "rate stability", i.e., in order for the user backlogs to be stable, the amount of buffered data during the scheduling period should not vary too much. The priority of UE n is thus computed as

$$P_n^R = \frac{T_n^{\alpha}}{R_n^{\beta}} \left(\frac{A_n}{R_n' \tau} \right)^{\gamma}, \tag{3}$$

where R_n' is the predicted average rate over the next τ amount of time, where τ is the time length for which the scheduling decision will be executed (i.e., m TTIs in the considered SPS). We refer to the SPS algorithm using the priority function defined in (3) as SPS-II.

The scheduler can be tuned by adjusting the values of α, β, and $0 \leq \gamma \leq 1$. Unlike the backlog-stable priority function above, for $\alpha = \beta = 0$ the UE's scheduling priority will depend on the relationship between the arriving traffic and the traffic that is expected to be served based on the predicted average rate.

V. Scheduling With Traffic Prediction

Network traffic prediction is known to be challenging, especially at the level of individual users over short periods of time, because the traffic characteristics emerge from a complex interaction of the user behavior and various application layer protocols. Motivated by recent works on robust adversarial learning, in [20] we propose a meta-learning scheme that enables the design of an adaptive traffic predictor. The meta-learning scheme consists of a set of sub-policies, each optimized to predict a particular kind of traffic, and of a master policy that is trained for choosing the best fit predictor dynamically based on recent prediction performance, using deep reinforcement learning.

Fig. 1 shows the meta-learning scheme, where a deep Q-network (DQN) agent is trained as the master policy and seven sub-policies are employed. Based on the internet traffic analysis reported in [19], we categorize network traffic into four types, namely long dynamic adaptive streaming over HTTP (DASH), short DASH, live video, and non-video. For each kind of traffic, we train a predictor using a long short-term memory (LSTM) neural network, and thus obtaining the first four sub-polices in the meta-learning scheme [21]. Besides that, we introduced three computationally inexpensive predictors: persistence model forecast (PMF), zero predictor (ZP), and average value predictor (AVP). PMF assumes the time series is persistent, and hence it uses the last observed value as the prediction. ZP outputs a constant value of zero and is expected to be used during idle periods. AVP outputs the average number of bytes that arrived during the last pre-defined period. Further details about the meta-learning scheme are provided in [20].

We evaluated the proposed meta-learning scheme on a variety of traffic traces consisting of video and non-video traffic. Our results show that it consistently achieves a high prediction accuracy despite changing traffic characteristics [20]. Therefore, in this study we integrate the predictor with the scheduler, as illustrated in Fig. 2, such that the predictions of the incoming traffic can be utilized to make proper scheduling decisions.

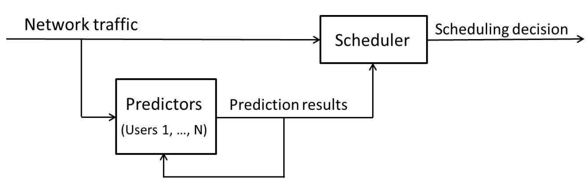


Figure 2. Block diagram of predictive semi-persistent scheduler.

VI. Performance Evaluation

In this section we evaluate the proposed scheduling algorithm through simulations. We first describe the simulation methodology, and then we provide numerical results.

A. Simulation Methodology

We consider a cellular network with 20 users uniformly distributed in its serving area, as shown in Fig. 3. The inter-site distance is 500 m. The network has two channels with 0.7 MHz bandwidth each. The transmit power for downlink is 30 dBm and the noise power is set to -100 dBm for all users. The channel gain is modeled as the composite effect of path loss with exponent 4, Rayleigh fading and log-normal shadowing with standard deviation of 8 dB [22].

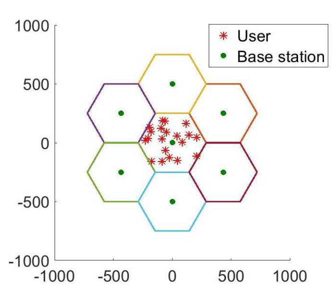


Figure 3. Cellular network topology.

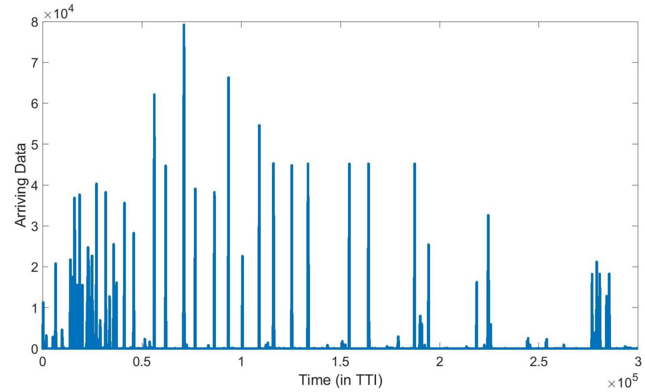


Figure 4. Data traffic per TTI arriving to User 1

In the considered system we set the TTI length to 1ms and calculate the scheduling solution over a period of 5 minutes (300000 TTIs). The initial backlog of each user is uniformly set to be 10000 bytes, and buffers are considered to be unlimited. The incoming traffic of each user is generated based on traffic traces collected using Wireshark, and consists of short video, long video, live video and web traffic. Thus, the simulation is trace-based. For most of the users the incoming traffic is dominated by video data and is thus highly bursty. As an example, we show the data traffic to User 1 in Fig. 4. The arriving traffic of the 20 users during the scheduling period ranges from 8.6 MB to 28.3 MB, with an average of 15.1 MB.

To evaluate the scheduling solutions, we consider user-level and network-level performance metrics. For user-level performance we calculate the per-user throughput, the per-user delay, and the number of occupied slots. For network-level performance we consider the sum throughput, the average of the per-user delay, channel utilization, and computational cost, which is defined as the number of scheduling decisions that are made by the scheduler during the given time period. In addition, we consider Jain's fairness index computed based on the number of active slots of each user (Fairness-I), and Jain's fairness index computed based on the per-user delay (Fairness-II). Simulations and result analyses are done using Python and Matlab.

B. Performance Evaluation

We apply SPS in the first channel and set $m = 100$ TTIs. In the second channel, a dynamic PF scheduler operating on a per TTI basis is employed. The reason of doing so is to achieve a computationally efficient scheduling solution and meanwhile to maintain a good service quality for users in fast changing environments. We test the proposed scheduling algorithms SPS-I (using the backlog-stable predictive priority function) and SPS-II (using the rate-stable predictive priority function), respectively. In both priority functions we set the parameters $\alpha = \beta = \gamma = 1$, and a latency window of $t_c = 100$. The incoming traffic is predicted by our proposed meta-learning scheme, as discussed in Section V. In order to evaluate how the prediction accuracy will impact the scheduling solution,

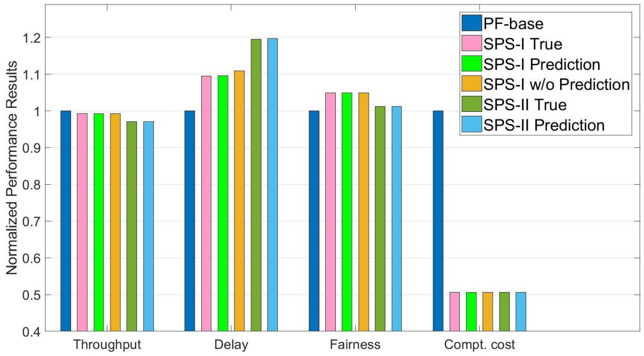


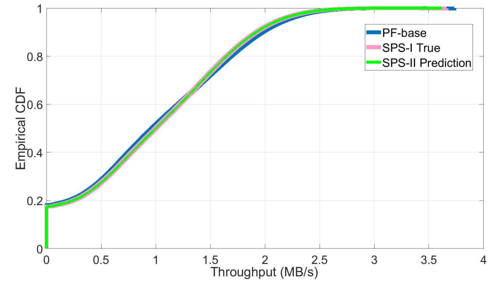
Figure 5. Network throughput, delay, fairness and computational cost for PF-base, SPS-I with true info, SPS-I with prediction, SPS-I w/o prediction, SPS-II with true info, and SPS-II with prediction.

we also run the test of the two scheduling algorithms with the true value of incoming traffic, which provides an upper bound of the scheduling performance. In addition, we test the case when SPS-I has no information of the arrivals, that is, in the priority function defined in (2), A_n is set to zero. We run all the tests and compare the scheduling solutions with the case where dynamic PF is used for both channels. Due to the page limit, in Table I we only show the network-level results of the two channels.

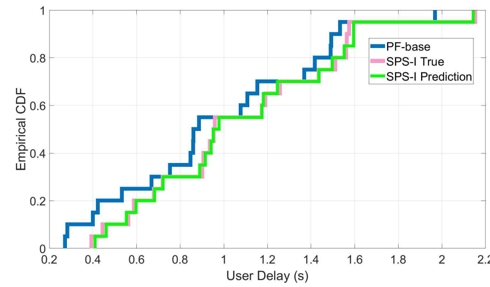
To ease comparison, we set the scheduling solution obtained by applying dynamic PF in both channels as the baseline (PF-base), and show the normalized results of the network-level throughput, delay, and fairness of the six schedulers in Fig. 5. Note that here we show results for Fairness-II, because users may have an empty buffer during the scheduling period due to the bursty nature of traffic, and hence comparing the number of active slots of each user may not lead to a reasonable result in terms of fairness. For the metric of delay, a higher value implies longer delay and hence worse performance.

Fig. 5 allows us to make a number of important observations. First, for the three classic performance metrics, i.e., throughput, delay, and fairness, both SPS-I with true and predicted traffic information achieve comparable performance as the baseline solution. Comparing to the two SPS-I schedulers, SPS-II schedulers perform worse in all three metrics, especially in the average delay. The observation implies that in the priority function the current buffer size B_n should be considered. In addition, in the rate-stable predictive priority function (3) the value of R_n' is estimated by the average historical rate, which may lead to inaccurate result if the channel conditions change fast. Secondly, we observe that the corresponding results of SPS-I without prediction information are clearly worse than those of SPS-I with prediction, indicating that traffic prediction could contribute to computing an effective scheduling solution.

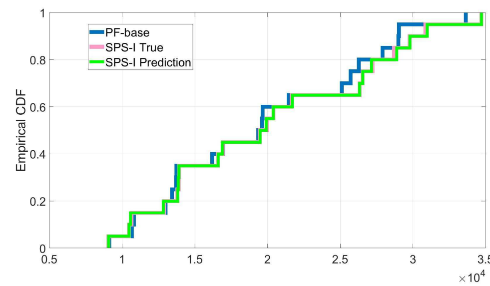
Next, we focus on the performance evaluation on SPS-I. To this end, we further investigate the results of SPS-I with true and predicted traffic information in Fig. 6, which shows the empirical cumulative distribution functions (CDFs) of the network throughput in each TTI, the delay of each user, and the number of occupied slots by each user.



(a) CDF of throughput in each TTI



(b) CDF of user delay



(c) CDF of number of occupied slots

Figure 6. Scheduling performance obtained using PF-base, SPS-True, and SPS-prediction.

Observing from Fig. 5 and Fig. 6, the throughput of PF-base is slightly better than that of the two SPS-I schedulers, but the difference is negligible. PF-base performs better than SPS-I with true or predicted traffic information in terms of delay, with a difference of less than 10%. Considering the fact that PF-base dynamically adjusts its decision in every TTI, while SPS-I is expected to do that in every 100 TTIs (unless the buffer of the active user becomes empty), the performance of SPS is rather satisfactory. Moreover, SPS provides more "fair" treatment to each user in transmission delay than the baseline solution does. Notable in Fig. 5 is the comparison of computational cost, by employing SPS in one channel, the computational cost of the two channels decreases almost by a factor of two, meaning that we can free up a considerable amount of computational resources for other purposes. The result of computational cost also gives information on the actual number of TTIs that SPS schedules. Using SPS-

Table I
Network Performance of dynamic PF, SPS-I with and w/o traffic information, and SPS-II with traffic information

Scheduler	Throughput (MB/s)	Delay (s)	Fairness-I	Fairness-II	Channel Utilization	Computational Cost
Dynamic PF	1.2164	1.0423	0.89	0.81	0.65	600000
SPS-I with true info	1.2074	1.1405	0.88	0.85	0.67	303751
SPS-I with prediction	1.2073	1.1423	0.88	0.85	0.67	303745
SPS-I w/o prediction	1.2072	1.1561	0.88	0.85	0.67	303780
SPS-II with true info	1.1809	1.2453	0.88	0.82	0.66	303765
SPS-II with prediction	1.1807	1.2472	0.88	0.82	0.66	303792

I with true traffic arrival information, the scheduler made 3751 scheduling decisions for 300000 TTIs, so the average persistence time of each scheduling decision is around 80 TTIs. For SPS with prediction, the average is almost the same, 80 TTIs. For the three CDFs shown in Fig. 6, we observe that the green curve (SPS-I Prediction) almost overlaps with the pink one (SPS-I True), implying that the proposed SPS scheme is reasonably resilient to noisy predictions. Overall, the results indicate that the proposed SPS-I performs well and it could be a promising approach for efficient downlink scheduling in cellular networks.

VII. Conclusions and Outlook

We have proposed a semi-persistent downlink scheduler integrating with an adaptive traffic predictor so as to efficiently compute an effective scheduling solution. We have proposed novel user priority functions that account for the predicted incoming traffic per user and the user buffer occupancy. We have evaluated the proposed scheduling algorithms by simulations. The numerical results show that the semi-persistent scheduler performs well at a significantly reduced computational cost in comparison to a dynamic scheduler. Overall, our results show that combined with computationally efficient traffic predictors, e.g., in tensor processing unit (TPU) hardware, prediction-aware semi-persistent scheduling could reduce the computational burden for downlink scheduling without affecting throughput and fairness.

We remark that both the scheduler and the predictor have high potential to be further improved or to be tailored for specific applications/services. Moreover, it would be of interest to investigate the optimal length that a scheduling decision should be persistent, and the number of subcarriers that should be used for SPS. An additional promising direction for future research could be to investigate scheduling schemes based on predicted channel conditions, while also considering the buffered data and the predicted data arrivals.

References

[1] G. Miao, J. Zander, K. W. Sung, and S. Ben Slimane, *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016.

[2] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in lte cellular networks: Key design issues and a survey," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 2, pp. 678–700, 2013.

[3] D. Jiang, H. Wang, E. Malkamaki, and E. Tuomaala, "Principle and performance of semi-persistent scheduling for VoIP in LTE system," in *Proc. of IEEE WiCOM*, Sep. 2007, pp. 2861–2864.

[4] A. Pantelidou and A. Ephremides, "Scheduling in wireless networks," *Found. Trends Netw.*, vol. 4, no. 4, pp. 421–511, 2011.

[5] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.

[6] R. Nelson and L. Kleinrock, "Spatial TDMA: a collision-free multihop channel access protocol," *IEEE Trans. Commun.*, vol. 33, no. 9, pp. 934–944, 1985.

[7] E. Arikan, "Some complexity results about packet radio networks," *IEEE Trans. Inf. Theory*, vol. 30, no. 4, pp. 681–685, 1984.

[8] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 2, pp. 166–177, 1993.

[9] C. G. Prohazka, "Decoupling link scheduling constraints in multihop packet radio networks," *IEEE Trans. Comput.*, vol. 38, no. 3, pp. 455–458, 1989.

[10] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[11] S. A. Borbash and A. Ephremides, "Wireless link scheduling with power control and SINR constraints," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5106–5111, 2006.

[12] P. Björklund, P. Värbrand, and D. Yuan, "A column generation method for spatial TDMA scheduling in ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 4, pp. 405–418, 2004.

[13] O. Goussevskaia, Y. A. Pswald, and R. Wattenhofer, "Complexity in geometric SINR," in *Proc. of ACM MobiHoc*, 2007, pp. 100–109.

[14] Y. Feng, A. Nirmalathas, and E. Wong, "A predictive semi-persistent scheduling scheme for low-latency applications in LTE and NR networks," in *Proc. of IEEE ICC*, May 2019, pp. 1–6.

[15] N. Afrin, J. Brown, and J. Y. Khan, "Design of a buffer and channel adaptive lte semi-persistent scheduler for M2M communications," in *Proc. of IEEE ICC*, June 2015, pp. 5821–5826.

[16] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proc. IEEE*, vol. 95, no. 12, pp. 2299–2313, Dec 2007.

[17] W. Jiang and H. D. Schotten, "Neural network-based channel prediction and its performance in multi-antenna systems," in *Proc. of IEEE VTC-Fall*, Aug 2018, pp. 1–6.

[18] V. Vukadinovic and G. Karlsson, "Video streaming in 3.5G: On throughput-delay performance of proportional fair scheduling," in *Proc. of IEEE MASCOTS*, 2006, pp. 393–400.

[19] "Cisco visual networking index: Forecast and trends, 2017–2022 white paper." [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[20] Q. He, A. Moayyedi, G. Dán, G. P. Koudouridis, and P. Tengkvist, "A meta-learning scheme for adaptive short-term network traffic prediction," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2271–2283, 2020.

[21] Q. He, G. P. Koudouridis, and G. Dán, "A comparison of machine and statistical time series learning for encrypted traffic prediction," in *Proc. of IEEE ICNC*, 2020, pp. 714–718.

[22] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.