A Multi-Level Game Theoretic Algorithm for Device-to-Device Resource Allocation with Frequency Reuse

Sebastian Lindner, Raphael Elsner and Andreas Timm-Giel
Institute of Communication Networks (ComNets)
Hamburg University of Technology (TUHH)
Hamburg, Germany
{sebastian.lindner, raphael.elsner, timm-giel}@tuhh.de

***Abstract*—Device-to-Device (D2D) communication as an extension to current mobile networks attracted a lot of attention during the last years. Along with introducing D2D communication as an underlay to cellular communication, radio frequency resources are proposed to be reused to increase spectral efficiency and system capacity. Being able to reuse frequency resources can lead to in-cell interference which must be actively mitigated. We investigate a novel joint radio resource scheduling and allocation algorithm which takes into account different user service classes and mitigates in-cell interference from frequency resource reuse. It is shown that system performance can be increased if resource allocation and reuse is done in a coordinated manner. The proposed scheduler is based on two algorithms from game theory, scales well due to its linear computational complexity and outperforms baseline schedulers in several simulated scenarios.**

***Index Terms*—Device-to-device communication, radio frequency reuse, game theory, radio resource allocation and scheduling**

I. Introduction

Worldwide mobile traffic is predicted to increase by 700 % within five years from 2016 until 2021, according to [1]. The authors of [2] formulate it as “mobile data sees exponential growth”. On the other hand, the width of the frequency bands licensed to Long Term Evolution (LTE) networks remains fixed, and so the increasing number of users compete for a fixed number of radio resource blocks (RBs), and researchers are in a race to improve upon communication standards so that the users can be accommodated in today’s networks.

A novel functionality is proposed through D2D, which the authors of [3] predict will be of major importance to fifth generation (5G) mobile systems. With D2D functionality, user terminals (UTs) can communicate to one another directly, skipping an intermediate hop to the eNodeB (eNB), the base station of an LTE cell. In [4] a number of scenarios are investigated that benefit from D2D functionality. For example, delay-critical car-to-car communications of urgent road conditions could be realized, or cell edges be softened through relay UTs.

Along with D2D the concept of frequency resource reuse (FRR) is proposed, according to [3]. With FRR a single RB can be used by more than one UT. In [4] we find that most researchers suggest *underlay inband D2D*, where the same frequency band licensed to LTE is used for D2D transmissions (inband), and no RBs are dedicated to D2D (underlay). Simultaneous transmission on the same frequency band inevitably generates in-cell interference. In [5] it is stated that downlink (DL) frequencies are more congested than uplink (UL) frequencies. Therefore UL RBs are proposed to be reused for D2D transmissions.

This allows us to define the optimization problem that resource scheduling and allocation in LTE poses: for each transmission time interval (TTI) it is asked which UTs are allocated which RBs, so that conflicting targets are maximized: 1) system throughput 2) user fairness 3) Quality of Service (QoS) provisioning.

A. Outline

In this work, a novel scheduling algorithm is proposed that combines two games from the field of game theory, to obtain a scheduler that is QoS-aware, considers estimated throughput and fairness among UTs, and is capable of finding UT pairs that are well-suited to share a RB through FRR. Section II covers related work. In Section III our model is described. Section IV gives a detailed description of the proposed algorithm. Section V contains the results we could obtain through simulation. Section VI concludes this document with a summary, conclusion and outlook.

II. Related Work

The authors of [6] use the Shapley value to find a fair distribution of the available RBs among UT coalitions, where UTs with similar application types are grouped together, e.g. the Voice-over-IP (VoIP) UTs form a coalition. The Shapley value distribution is “fair” in the sense that UT coalitions that require more RBs to satisfy all member UT demands are assigned more. In a second step the coalition RBs are allocated to coalition member UTs through the Exponential / Proportional Fair (EXP/PF) rule, which differentiates between real time (RT) and non-RT UTs and considers expected throughput and fairness for non-RT UTs, and additionally packet delay for RT UTs. We use this approach as the basis for our scheduler.

Stackelberg games are played between all (cellular UT, D2D UT pair)-pairs in the scheduler proposed in [7]. Cellular UTs are assumed to have been allocated RBs already, and D2D UT transmitters are supplied with RBs through Stackelberg games. In each gameplay the D2D UT transmission power is determined and the expected interference from the frequency reuse of the cellular UT's RB is calculated, forming a scheduling priority. Cellular UT RBs are reused according to the best-matched pairs, so those with the largest priority. This will be the second building block for our scheduler.

Furthermore, the concept of Stackelberg games is used by the authors of [8] in a D2D relay scenario. They improve cell-edge user performance by using an amplify and forward relaying scheme. By using a two-level game they find appropriate relay pairs and set optimal transmission powers to mitigate interference introduced by frequency reuse.

An extensive overview on different game theoretical approaches for D2D resource allocation is given in [9]. The authors categorize and summarize possible games for different scenarios and use cases. A *non-cooperative power control game* can model the interests of cellular and D2D UTs involved in FRR. In a self-organized manner, D2D UTs update their transmission powers to mitigate interference. A *reverse iterative combinatorial auction* lets cellular and D2D UTs be auctioneers of RB packages, and the cellular network places bids to obtain demands from UTs.

The authors of [10] define an interference aware graph where vertices correspond to UTs and edges between two vertices hold weights corresponding to the *potential mutual interference* between them. Every UT sorts the RBs by descending estimated signal to interference and noise ratio (SINR), leading to RB clusters containing those UTs that are interested in it. In an iterative procedure the sum of channel capacities of each link in the network is maximized, and so a solution allocation is found that allows FRR and explicitly considers the generated in-cell interference.

III. Model Description

We investigate a single cell LTE scenario. D2D and FRR are enabled, and RBs are scheduled by the eNB. We assume a configuration of 6 UL RBs and 6 DL RBs available each TTI. Especially video streaming, VoIP and full buffer UTs are considered. UTs form pairs, where cellular pairs communicate to one another through the eNB, while D2D UTs communicate directly using UL RBs. For pathloss and fading models, an urban macrocell model and Jakes fading are used as implemented in the simuLTE framework, found in [11], for the OMNeT++ simulator [12]. The thermal noise power density is set to $-174\,\mathrm{dBm/Hz}$. $95\,\%$ confidence intervals are given. We compare baseline scheduling algorithms Round Robin (RR) and Proportional Fair (PF) to the proposed scheduler described in Section IV.

IV. Algorithm Description

The scheduling algorithm we propose builds upon two base schedulers: the Transferable Utility (TU) scheduler from [6] is combined with the Stackelberg scheduler from [7]. The latter is modified in ways explained in Section IV-B1. The proposed scheduler is summarized in Algorithm 3.

A. Transferable Utility game-based scheduler

In [6] a cooperative TU game is used as a building block for a QoS-aware scheduler. In two steps, first UT coalitions are identified and supplied with RBs using the Shapley value, in accordance with the RB demand each coalition reports. Using the EXP/PF rule, RBs are allocated to individual UTs inside a coalition.

1) First step: fair division of RBs among UT coalitions: The authors of [6] group UTs into coalitions that run similar applications; in their work, VoIP, video streaming and constant bit-rate coalitions are investigated, making up the coalitions C. A *bankruptcy game* is modeled where a resource $E = \{\text{all RBs for one TTI}\}$ is shared among the players $C = \{\text{all UT coalitions}\}$, but the aggregate demand $\sum_{i\in C} d_i$ exceeds the number of RBs available $|E|$, hence the name. A TU game requires a *characteristic function* $v : 2^N \to \mathbb{R}$ that determines the utility of each coalition $c \in C$. According to [13], a bankruptcy game can be converted into a TU game using Equation 1. For coalition c the characteristic function returns the resources that remain after all users not in c have received their demands.

$$v(c) = \max\{0,\, E - \sum_{i\notin c} d_i\} \tag{1}$$

The obtained TU game $(N,\, v)$ allows us to fairly distribute E using the Shapley value. We can order our $n = |C|$ coalitions in $n!$ ways. A particular ordering is $\pi \in \Pi$. Going through π sequentially, the *marginal contribution* of each coalition $c_i \in \pi$ is $v(p^i_\pi \cup i) - v(p^i_\pi)$, where $p^i_\pi = (1, 2, \dots, i-1)$ denotes all coalitions preceding i in ordering π. The marginal contributions depend on the specific ordering π it is evaluated for, and since in Equation 1 v is defined as "what resources are left for me after all other coalitions have been served", the coalitions' utilities are highly dependent on the order they are served. The Shapley value evaluates these contributions for all orderings, and the average over these makes up the Shapley values for coalitions $c \in C$ as in Equation 2.

$$\phi_c(v) = \frac{1}{n!}\sum_{\pi\in\Pi}\left[v(p^i_\pi \cup i) - v(p^i_\pi)\right] \tag{2}$$

To apply this to our problem, the respective coalition demands d_c are determined, the TU game formulated, and the Shapley value shares ϕ_c determined, which correspond to the number of RBs that should be allocated to coalition c, taking into account the coalition demand d_c, so those that require more RBs are allocated more.

2) Second step: EXP/PF rule resource allocation: Having scheduled RB pools to UT coalitions, the individual UTs tap the respective RB pools using the EXP/PF rule. In TTI t UT j out of N is selected for the next RB according to Equation 3.

$$j = \max_{i \in N} \begin{cases} \exp\left(\frac{a_i V_i(t)}{1+\sqrt{\overline{V_i}}}\right) \frac{\mu_i(t)}{\overline{\mu}_i} & \text{if } i \text{ is a real-time flow,} \\ \frac{\mu_i(t)}{\overline{\mu}_i} & \text{if } i \text{ is a non-real-time flow.} \end{cases} \tag{3}$$

$\mu_i(t)$ denotes the estimated data rate of UT i in TTI t, $\overline{\mu}_i$ is the past average throughput, and $\mu_i(t)/\overline{\mu}_i$ is the PF metric. $V_i(t)$ is the packet delay that UT i is experiencing, $\overline{V_i(t)}$ is the past average packet delay and a_i corresponds to the UT's packet delay target. For non-RT flows the PF metric is applied, and for RT flows the metric is augmented by an exponential function representing packet delay. The distinction between RT and non-RT makes the scheduler QoS-aware as it considers packet delay as a key performance indicator (KPI) for RT users. This leads to Algorithm 1.

Algorithm 1 Transferable Utility game-based scheduler

procedure SCHEDULE_TU(active users N, RBs R)
  $C \leftarrow \{C_{\text{CBR}} = \{\}, C_{\text{VoIP}} = \{\}, C_{\text{vid}} = \{\}\}$
  $C \leftarrow$ **updateClasses**(N, C_{CBR}, C_{VoIP}, C_{vid})
  $d_{\text{vid}}, d_{\text{VoIP}}, d_{\text{CBR}} \leftarrow$ aggregate class RB demands
  $\phi \leftarrow$ **shapley**(C_{CBR}, d_{CBR}, C_{VoIP}, d_{VoIP}, C_{vid}, d_{vid}, R)
  for $r_i \in R$ **do**
    $j \leftarrow$ **EXP_PF**(C_{CBR}, C_{VoIP}, C_{vid}, r_i, p)
    schedule r_i to j
  end for
end procedure

B. Stackelberg game-based scheduler

In a Stackelberg game as modeled in [7], a cellular *leader* k UT plays against a D2D UT pair *follower* i. The leader "owns" a RB, and the follower wishes to reuse it through FRR. The leader demands a *reuse fee* α_k from the follower for reusing their RB. The first step of the game is the leader setting α_k. The follower reacts to α_k by setting their own transmission power p_i. Both players have utility functions u_k and u_i which they aim to optimize through their respective moves. The leader dominates the game as they can exploit the leader's advantages: 1) they can act first 2) they know u_i 3) they know the follower will pick a *best response*, i.e. that p_i that maximizes u_i. Through *backward induction* optimal strategies are found that provide the scheduling algorithm.

$$u_i(\alpha_k, p_i) = \log_2\left(1 + \frac{p_i g_{ii}}{p_k g_{ki} + N_0}\right) - \alpha_k p_i g_{ie} \tag{4}$$

u_i in Equation 4 consists of the estimated channel throughput from the *Shannon-Hartley theorem* minus the α_k-weighted interference for the leader. p_i, p_k are the two player's transmission powers, N_0 the noise constant, and the channel gain between leader and follower is captured in g_{ki}, between the D2D follower pair in g_{ii} and between follower and eNB in g_{ie}. $p_i g_{ie}$ is therefore the power from the follower's transmission that arrives at the eNB, which is the destination of the cellular leader's transmission, and therefore the interference for the leader. As the interference term grows linearly with p_i, while the channel throughput grows logarithmically, there is an optimal $\hat{p}_i$ the follower will pick. α_k changes the slope of u_i, so, foreseeing the $\hat{p}_i$ that optimizes a respective u_i given a particular α_k, the leader searches that α_k that forces the follower to pick a specific $\hat{p}_i$ that in turn *benefits the leader the most*. Therefore the interference for the leader is actively managed by forcing the follower's transmission power. As u_i also corresponds to the scheduling priority of this (leader, follower)-pair, if the follower is not satisfied with the outcome, the scheduling priority is going to be small and another pair favored instead.

1) Modifications: The authors of [7] assume that leaders are cellular UTs and followers are D2D UTs. We propose, however, that a scheduler should be *general* in the sense that it should work with arbitrary numbers of cellular and D2D UTs, so that both D2D-dense and -sparse networks can be served. If either number becomes comparatively large or small the scheduler in [7] would unnecessarily neglect one UT type, or may stop working altogether if there are no cellular UTs that could share their RBs with D2D UTs. We therefore modify the scheduler by incorporating two *mappings* that allow not only (cellular, D2D)-pairs, but also (D2D, cellular)- and (D2D, D2D)-pairs to play Stackelberg games. We identify g_{ke} as the leader's *useful* channel gain as it describes the channel condition between the leader and their destination, the eNB. For an inverse game, when the leader is not of cellular, but of D2D type, the destination changes to a D2D receiver, and so we write a new channel gain $g_{k_1 k_2}$ into g_{ke}, so that the variable still represents the same idea – the useful channel's condition of the leader. Likewise we write a new g_{ik_2} into g_{ie} so that the interfering channel from the leader's perspective is modeled. This explains the m_{inverse} mapping in Equation 5.

$$\begin{aligned} m_{\text{inverse}} = \{&g_{ke} \leftarrow g_{k_1 k_2}(\text{new}),\, g_{ki} \leftarrow g_{ke}, \\ &g_{ii} \leftarrow g_{ie},\, g_{ie} \leftarrow g_{ik_2}(\text{new})\} \end{aligned} \tag{5}$$

In an analog way m_{D2D} is found in Equation 6.

$$\begin{aligned} m_{\text{D2D}} = \{&g_{ke} \leftarrow g_{k_1 k_2}(\text{new}),\, g_{ki} \text{ remains}, \\ &g_{ii} \text{ remains},\, g_{ie} \leftarrow g_{ik_2}(\text{new})\} \end{aligned} \tag{6}$$

When necessary a mapping is applied and so we could modify the algorithm from [7] to be more general and allow all but (cellular, cellular)-pairs to play. This explains Algorithm 2.

Finally our proposed scheduler combines algorithms 1 and 2 into Algorithm 3. All users are scheduled in a QoS-aware manner. Those that were scheduled RBs are named leaders, and those that received no RBs are named followers. Followers now play Stackelberg games with the leaders, and the best-matched pairs share the leader's RBs through FRR. Figure 1 gives a graphical description of Algorithm 3.

V. Simulation Results

The proposed scheduler is implemented as a scheduling module for the simuLTE LTE simulation framework from [11].

Algorithm 2 Modified Stackelberg Scheduler

procedure SCHEDULE_MODDED_ST(leaders L, followers F, RBs R)
  initialize priority queue $P_{i,k}$
  for $l \in L$ **do**
    for $f \in F$ **do**
      if l is cellular **and** f is cellular **then**
        continue
      end if
      if l is cellular **and** f is D2D **then**
        pair $p \leftarrow (l, f)$
      end if
      if l is D2D **and** f is cellular **then**
        pair $p \leftarrow m_{\text{inverse}}(l, f)$
      end if
      if l is D2D **and** f is D2D **then**
        pair $p \leftarrow m_{\text{D2D}}(l, f)$
      end if
      $P_{l,f} \leftarrow$ Stackelberg game result for pair p
    end for
  end for
  sort P in descending order
  while $|P| > 0$ **do**
    $(l, f) \leftarrow P$.pop()
    if !scheduled(l) **and** !scheduled(f) **then**
      schedule l's resources to f
      set f's transmission power to resp. $\hat{p}_i$
    end if
  end while
end procedure

Algorithm 3 Proposed Scheduler

procedure SCHEDULE(active users N, RBs R)
  $map_{\text{leaders}} \leftarrow$ **schedule_tu**(N, R)
  $N_{\text{leader}} \leftarrow \{\}$, $N_{\text{follower}} \leftarrow \{\}$
  for $n \in N$ **do**
    if $|map_{\text{leaders}}[n]| > 0$ **then**
      $N_{\text{leader}} \leftarrow \{N_{\text{leader}}, n\}$
    else
      $N_{\text{follower}} \leftarrow \{N_{\text{follower}}, n\}$
    end if
  end for
  $map_{\text{followers}} \leftarrow$ **schedule_modded_st**($N_{\text{leader}}, N_{\text{follower}}$)
  for $r \in R$ **do**
    for $n \in N$ **do**
      if $r \in map_{\text{leaders}}[n]$ **or** $r \in map_{\text{followers}}[n]$ **then**
        schedule r to n
      end if
    end for
  end for
end procedure

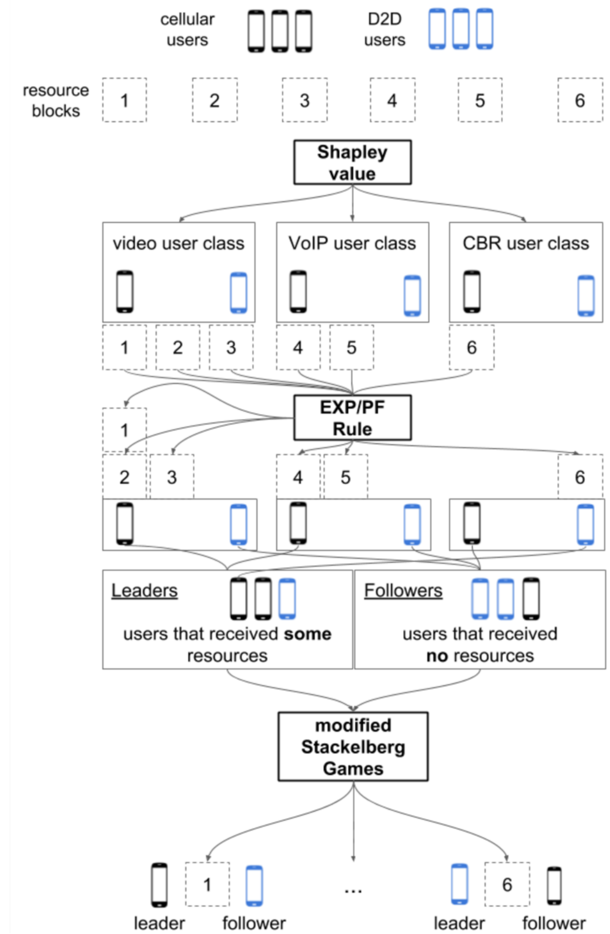
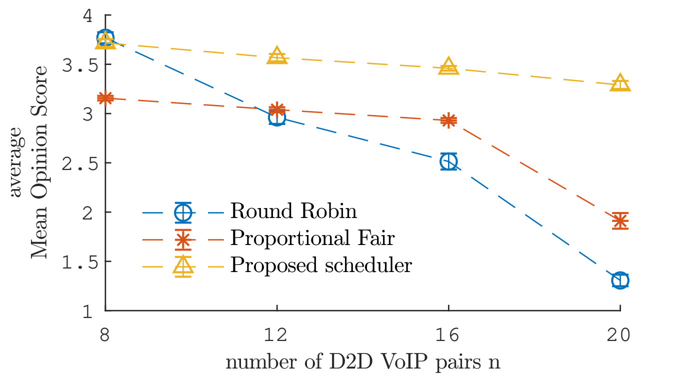


Fig. 1. Overview of the proposed Algorithm 3.

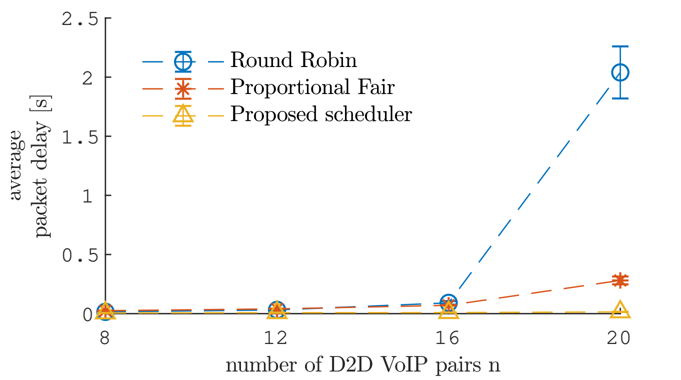
A. Voice-over-IP scenario

In one scenario the effectiveness of considering packet delay in RT applications such as VoIP through the EXP/PF rule is investigated. A number $n = 8, 12, 16, 20$ of D2D VoIP UT pairs compete for the 6 RBs. They are randomly positioned so they are at most 46 m from the next UT. According to [14], VoIP performance can be captured in the Mean Opinion Score (MOS), which ranges from 5 (excellent) to 1 (bad). We simulate for 15 s and repeat each simulation 30 times. Figure 2 depicts the simulation results, where Figure 2a shows the average MOS of all UTs, while Figure 2b shows the average packet delay. When the number of RBs is not much smaller than the number of users n, it suffices to sequentially schedule RBs to users with RR. PF compensates bad channel conditions with more RBs, but for VoIP throughput is not a KPI as the voice data is usually small, and so this compensation leads to a worse MOS. As n grows, RR's sequential scheduling leads to a skyrocketing packet delay and an abysmal MOS. PF is substantially more robust for large n, but the MOS suffers nonetheless as $n = 20$. The robustness can be especially observed at $n = 20$ where the 95 %-confidence interval for RR is much larger than that of PF.

The proposed scheduler is by far the most robust of the



(a) VoIP UTs average Mean Opinion Score over an increasing number of UT pairs n.



(b) VoIP UTs average packet delay over an increasing number of UT pairs n.

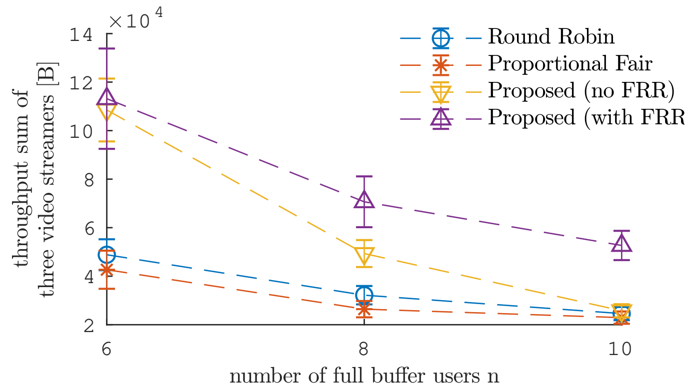
Fig. 2. Comparison of three schedulers in a D2D VoIP scenario. VoIP performance is evaluated as the Mean Opinion Score in 2a. The average packet delay is shown in 2b, which influences the Mean Opinion Score.

tested schedulers. It can be seen in Figure 2b how the EXP/PF rule keeps packet delay small even for $n = 20$ and this is reflected in a MOS that decreases only slightly as $n \to 20$.

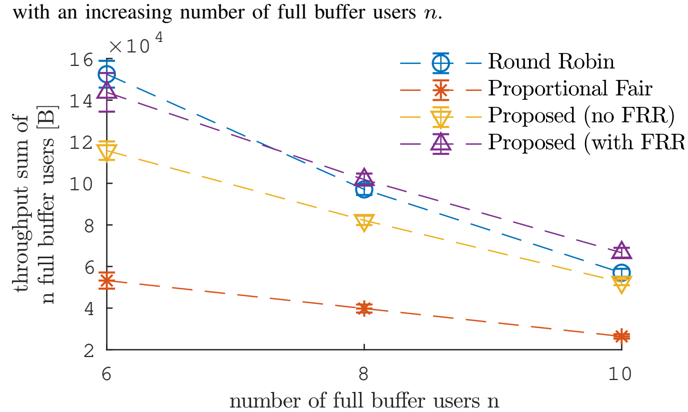
B. Video streaming and full buffer scenario

In this scenario a fixed number of three video streaming D2D UT pairs compete with an increasing number of $n = 6, 8, 10$ full buffer D2D pairs for the six available RBs per TTI. The three video streaming UT pairs' transmitters and receivers are positioned $40, 50, 60\,\text{m}$ from each other, so that the pairs have increasingly bad channel conditions. Full buffer UT pairs are randomly positioned with an intra-pair distance of at most $46\,\text{m}$. We simulate for $10\,\text{s}$ and repeat the simulation 15 times. Figures 3a and 3b show the sum throughput of the three resp. n user pairs.

Figure 3b shows that full buffer UTs benefit from RR not trying to compensate worse channel conditions of video streaming UTs. As there are more full buffer UTs and since RR sequentially schedules RBs to UTs, full buffer UTs have the largest sum throughput with RR. Consequently *Jain's fairness index* J, from [15], which ranges between $J = 1$ (best) and $J = 1/n$ (worst), is computed and found to be comparatively bad at $J \approx 0.85$. PF compensates worse channel conditions with more RBs scheduled to such UTs, lowering



(a) Throughput sum of three realtime video streaming UTs as they share RBs with an increasing number of full buffer users n.



(b) Throughput sum of n non-realtime full buffer users as they share RBs with three video streaming users.

Fig. 3. Three video streaming UTs compete with an increasing number of full buffer UTs n for the available RBs. Throughput sum of the three video streaming UTs is shown in 3a. Throughput sum of the n full buffer UTs is shown in 3b.

sum throughput, but lifting fairness to $J \approx 0.95$ for video streaming UTs and $J \approx 0.975$ for full buffer UTs.

The proposed novel scheduler is equally fair. In Figure 3a we see that it also prioritizes the RT video streaming UTs and increases their sum throughput at the cost of non-RT full buffer UTs performance. Video streaming UTs report a throughput sum only slightly smaller than the throughput sum of n full buffer UTs. Enabling FRR could be shown to increase system capacity as both user types report a larger sum throughput. In another test we could show that randomly picking UT pairs for FRR can substantially decrease system performance due to in-cell interference, consolidating that an intelligent interference management is crucial for a worthwhile introduction of FRR.

VI. Summary, Conclusion, and Outlook

A. Summary

We have recognized two schedulers from [6] and [7] that both make use of game theory as complementing each other. Both have linear computational complexity and the proposed scheduler remains linear. The first scheduler, described in Section IV-A, uses the Shapley value and the EXP/PF rule to schedule RBs in a QoS-aware manner. The second scheduler, described in Section IV-B, finds UT pairs that are well-suited to share a RB in one TTI through playing Stackelberg games. The authors in [7] had not considered the initial scheduling of RBs and instead focused on this pair finding. We therefore picked the TU-based scheduler as a good match to take on this task. Through the modifications in Section IV-B1 to the Stackelberg-based algorithm, we could eliminate the second downside. Originally only cellular UTs could be scheduled RBs in a first step, and D2D UTs were provided with RBs through FRR. This would mean that only cellular UTs could benefit from a QoS-aware scheduler, neglecting D2D UTs. Likewise, the scheduler would not have been general in the sense that if only D2D UTs are present in a network, the scheduler could not work. Through our modifications both UT types can benefit from the QoS-aware TU-based scheduler, and the scheduler is now general.

B. Conclusion

We could show that the proposed scheduler outperforms the baseline schedulers Round Robin and Proportional Fair in a VoIP scenario and in a mixed video streaming and full buffer scenario. The Stackelberg games successfully manage interference, which we could see by comparing system performance to a scheduler that randomly picks UT pairs for frequency resource reuse in another test. Frequency resource reuse could be shown to increase system capacity when interference is managed. The Quality of Service-aware Transferable Utility-based scheduler lets real time applications benefit especially as they are prioritized. Non-real time applications see a decreased performance in return, which is a welcome behaviour as these can be expected to cope with a decreased performance more easily. Regarding fairness among UTs the proposed scheduler is as fair as PF, and more fair than RR, which we could see from calculating Jain's fairness indices. Through the modifications to the Stackelberg-based games we could achieve fairness among cellular and D2D UTs as well, which we could observe in other tests.

C. Outlook

We could witness unnecessary performance degradation in cellular UTs in specific scenarios which had not been present prior to the modifications to the Stackelberg games, and this problem remains to be investigated.

Also, some doubt can be raised concerning the choice of the Shapley value for the initial distribution of RBs to UT coalitions. The Shapley value considers the RB demand of each coalition. This means that low-bandwidth applications such as VoIP are neglected, which will report a small RB demand, but may need more RBs to meet their packet delay target. An approach would be the modification of this RB demand to include a packet delay target as well.

We could only compare the proposed scheduler to baseline schedulers, showing how much better it does compared to a lower limit. It would be interesting to see how it performs compared to optimal scheduling, so to the upper limit.

The Stackelberg-based scheduler lacks, like the Shapley value method, any notion of QoS-awareness and focuses solely on finding well-matched FRR UT pairs. The follower utility function could be augmented to include the packet delay for RT applications like the EXP/PF rule does.

The Stackelberg-based scheduler is currently an oracle-based scheduler as it requires information about the channel conditions between UTs that the eNB can not be expected to know. To obtain these channel conditions a signaling protocol would have to be devised, and the effectiveness of the entire approach then needs to be investigated as the overhead might not be worth the effort.

To sum it up, the proposed scheduler performs satisfactorily and improves significantly upon simpler schedulers, while not requiring prohibitively large computational costs. Looking into the open research questions should help in further enhancing the scheduling performance.

References

[1] C. Systems, "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper." [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[2] S. A. Alqahtani and M. Alhassany, "Comparing different LTE scheduling schemes," in *2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013*. Shanghai, China: IEEE, 2013, pp. 264–269.

[3] S. Yu, W. Ejaz, L. Guan, and A. Anpalagan, "Resource Allocation Schemes in D2D Communications: Overview, Classification, and Challenges," *Wireless Personal Communications*, vol. 1, pp. 1–20, 2017.

[4] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communcation in Cellular Networks," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 4, pp. 1801–1819, 2014.

[5] R. Alkurd, R. M. Shubair, and I. Abualhaol, "Survey on device-to-device communications: Challenges and design issues," in *2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS)*. Trois-Rivières, Canada: IEEE, 2014, pp. 361–364. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6934057

[6] M. Iturralde, A. Wei, T. Yahiyda, and A.-L. Beylot, "Resource allocation for real time services using cooperative game theory and a virtual token mechanism in lte networks," in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 2012.

[7] Feiran Wang, L. Song, Z. Han, Q. Zhao, and X. Wang, "Joint scheduling and resource allocation for device-to-device underlay communication," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, apr 2013, pp. 134–139. [Online]. Available: http://ieeexplore.ieee.org/document/6554552/

[8] J. Zhao, K. K. Chai, Y. Chen, J. Schormans, and J. Alonso-Zarate, "Two-level game for relay-based throughput enhancement via D2D communications in LTE networks," *2016 IEEE International Conference on Communications, ICC 2016*, pp. 1–6, 2016.

[9] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 136–144, jun 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6845058/

[10] R. Zhang, X. Cheng, L. Yang, and B. Jiao, "Interference-Aware Graph Based Resource Sharing for Device-to-Device Communications Underlaying Cellular Networks," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. Shanghai, China: IEEE, 2013.

[11] "simuLTE." [Online]. Available: http://simulte.com/

[12] "OMNeT++." [Online]. Available: https://omnetpp.org/

[13] J. González-Díaz, I. García-Jurado, and M. G. Fiestras-Janeiro, *An Introductory Course on Mathematical Game Theory*. American Mathematical Soc., 2010.

[14] L. Goodman, "Service level agreements based on objective voice quality testing for voice over IP (VOIP) networks," 2007.

[15] R. Jain, *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Wiley, 1991, vol. 491.