Impact of Time in Network Selection for Mobile Nodes

Tobias Rueckelt
Adam Opel AG
Bahnhofsplatz
Rüsselsheim, Germany
tobias.rueckelt@opel.com

Daniel Burgstahler
Technische Universität Darmstadt
Rundeturmstr. 10
Darmstadt, Germany
daniel.burgstahler@kom.tu-darmstadt.de

Florian Jomrich
Adam Opel AG
Bahnhofsplatz
Rüsselsheim, Germany
florian.jomrich@opel.com

Doreen Böhnstedt
Technische Universität Darmstadt
Rundeturmstr. 10
Darmstadt, Germany
doreen.boehnstedt@kom.tu-darmstadt.de

Ralf Steinmetz
Technische Universität Darmstadt
Rundeturmstr. 10
Darmstadt, Germany
ralf.steinmetz@kom.tu-darmstadt.de

ABSTRACT

Today, mobile nodes use multiple Internet access networks inefficiently. State-of-the-art network selection strategies distribute data traffic to available networks, but ignore an important second dimension: time. Time selection offers the opportunity to plan usage of future-available networks for delay-tolerant data traffic. We hypothesize, that concurrent selection of network and time leads to synergy effects, which reduce transmission cost and boost connectivity performance. To assess data distribution to wireless networks and time, we propose a novel rating model for joint network and time selection. The proposed model rates the satisfaction of Quality-of-Service (QoS) application requirements and trades off conflicting optimization goals. Moreover, we analyze the impact of time in network selection and present three network selection schedulers, which differ in their time selection strategy. Evaluation of the results reveals a strong impact of time selection on network performance. This gives evidence, that our initial hypothesis holds and forward-looking scheduling strategies provide a substantial benefit over state-of-the-art approaches.

Keywords

Network selection, time selection, rating model, mobile node

1. INTRODUCTION

Mobile devices are commonly equipped with multiple network interfaces, e.g. WiFi and cellular network. A method to switch and use them in parallel is provided by multi-homing capable handover schemes, which enable dynamic distribution of data flows over available networks [9]. This leads to the question: Which networks suit best to transmit application data?

MSWiM '16 Malta

 ISBN 123-4567-24-567/08/06. . . $15.00
DOI: 10.475/123_4

Network selection ideally aims on QoS requirement satisfaction of data flows. These QoS requirements are defined as flow-specific limits whose violation causes user-perceived service quality degradation. QoS requirement satisfaction benefits from flow distribution through qualified network selection. Networks are selected that match best to application QoS requirements. This contributes to our superordinate goal: user-perceived network quality improvement [8].

Considering a scenario of fast moving mobile nodes, network availability and transmission characteristics change frequently. Such dynamics impose challenges but also provide opportunities for substantial improvement. The question of network selection expands in time dimension: Which of the now or future-available networks suit best to transmit application data?

However, related work limits the problem to pure network selection and ignore the time selection. In contrast, the selection of proper transmission time without a view on network selection is often named resource allocation. Existing approaches usually focus on a single network connection and homogeneous data traffic. However, even these simple scenarios cover a huge optimization potential. Transmission of application data is coordinated to preload data or to use prospective available network resources. However, state-of-the-art QoS rating functions on resource allocation are not flexible enough for heterogeneous data traffic. Furthermore, no parallel networks are considered.

To the best of our knowledge, network selection and resource allocation are only considered separately in existing approaches. We hypothesize, that time selection in network selection has a strong impact on user-perceived network performance. To illustrate the impact of the time dimension, we give an example: Data transmission of a flow may be delayed to use a better-suited network, which will be available soon. Furthermore, delaying transmission or handover to another network can free up resources to finish concurrent data flow transmission before a certain network gets out of reach. Consequently, the scenario requires tight coordination of data flows in network and time selection.

To answer our hypothesis, we need a schedule rating function that enables assessment for joint network and time selection schedulers. In this paper, we present three contributions:

1. A rating model to assess schedule quality in joint network

and time selection. It considers satisfaction of application QoS requirements and monetary cost in heterogeneous networks with heterogeneous data traffic.

2. An analysis on the impact of time in network selection. Therefore, we present three schedulers differing in time impact of their search strategy. (1) A classical Network Selection (NS) without time impact. For each data flow, it selects the best matching now-available network. (2) An Opportunistic Network Selection (ONS) with statistical time impact. It extends NS by the opportunity to delay transmission, if no well-matching network is available now. (3) A Transmission Planning (TP) with explicit time impact on search strategy. It selects the best matching network from now and future available ones and decides about transmission time in a second step. Using these three schedulers and the proposed rating model, we investigate the impact of time selection in network selection schedulers on user-perceived network performance.

3. We provide the code sources of the schedulers and our evaluation framework including the novel rating model to encourage future development of advanced schedulers.[1]

In the following section, we discuss related work and give an overview on our schedule rating approach in section 3. Subsequently, we explain the parameter space in section 4 and the rating model in section 5. These sections belong to our first contribution, the schedule rating model. The following sections focus on our second contribution, the investigation of the impact of time in network selection. To demonstrate the joint network and time selection problem complexity, we analyze it in section 6 and discuss why existing heuristics are not applicable. After that, we present three heuristic schedulers in section 7. Finally, we apply our novel rating scheme to them in section 8 to investigate the impact of time in network selection.

2. RELATED WORK

The related work in this topic is split into two domains: network selection and resource allocation. In fusion, we call it scheduling. This split preempts work from evolving synergy benefits.

2.1 Network Selection

Network selection algorithms try to select the best network to connect to and a good point in time for a soft or hard handover. Mathematical models are analyzed in [12]. We further divide the related work into network-controlled and client-controlled selection. The difference is firstly the location of the decision making and triggering and secondly the environmental knowledge that decisions are based on.

Network-controlled: In the network-controlled area, network operators manage selection. In general, network operators aim to maximize overall throughput and their revenue. Network-controlled approaches benefit from top-level knowledge about user devices, which enables access coordination inside the controlled network. Competitive and collaborative design, using game theory [11] or predictive models are preferred and lead to remarkable results [6].

Client-controlled: In contrast, client-controlled approaches benefit from network operator independence. The network selection is not limited to those of the network operator. The selection uses detailed information about the user's needs. However, in the client-controlled domain, there is a lack of information about other users and the current network state. Therefore, algorithms are usually based on assessment of estimated network performance [13]. Many approaches use map-attached averaged historical data and movement estimation to derive future network availability and performance. The assessment measure in several papers is satisfaction of application QoS requirements. Popular methods are Multi-Attribute Decision Making (MADM) functions [7, 1], which apply weighted sums to assess the networks. MADM approaches provide linear models, which enable linear replacement of violations. However, linear replacement implies, that a strong violation of one requirement, is as severe as many slight or even negligible violations. Furthermore, normalization of requirements leads to fairness between data flows. However, a data flow with hard requirements should not be assessed equally to one with soft requirements, e.g. a voice-over-IP flow in comparison to a background download.

[1] http://www.kom.tu-darmstadt.de/~trueckelt/scheduling/

2.2 Resource Allocation

Resource allocation decides at which rate and which time data is transmitted. It focuses on collaborative link capacity use and packet preference. It is usually applied on data link layer and focuses on fairness of transmission. A well-studied subject is packet prioritization in WiFi within send buffers, to optimize transmission via a single link [4]. In contrast to this, we target collaborative use of different network interfaces. A simple L2 approach distributes packets to available networks in round robin fashion [14]. It reduces QoS requirements to interface preference and ensures fairness for scheduled flows. However, this scheduling covers only short-term delivery plans for the next few packets. The rating model is static and does not cover long-term optimization potential.

Bui and Widmer focus on bufferable video delivery [2]. They reduce device energy consumption and avoid buffer under-run. Therefore, flow control is planed for up to several minutes. Moreover, the scheduling tolerates hazy network prediction. This proactive long term scheduling provides good transmission time planning. A similar approach using opportunistic algorithms has been studied in [5]. However, authors of both papers limit their approaches to a single network and bufferable video traffic. Its QoS requirements are unchallenging and therefore mostly neglected. Thus, their schedule quality rating models lose general validity.

2.3 Discussion

In network selection, network-controlled and client-controlled schemes are often discussed as competitive approaches. In contrast, we argue that both should be applied concurrently. This way, network operators optimize performance with collaborative client information for their own network. Concurrently, users choose between optimized networks of different providers. For our scheduling, we assume this two-fold network selection model and focus on the user-controlled one.

Network selection and resource allocation both suffer from issues in rating functions. Those functions often lack validity for heterogeneous data traffic or provide only linear models. Linear models make QoS requirement violations interchangeable, regardless of their strength. A better model would avoid severe violations.

In addition, network selection and resource allocation provide only limited optimization potentials when regarded separately. Synergies evolve in a combined approach that plans both together. To assess such scheduling models and to analyze synergy effects, a combined rating model is required.

3. RATING MODEL OVERVIEW

Rating of schedules is fundamental for scheduler development. Therefore, we present a novel, well-elaborated metric that supports detailed schedule assessment for heterogeneous data traffic in joint

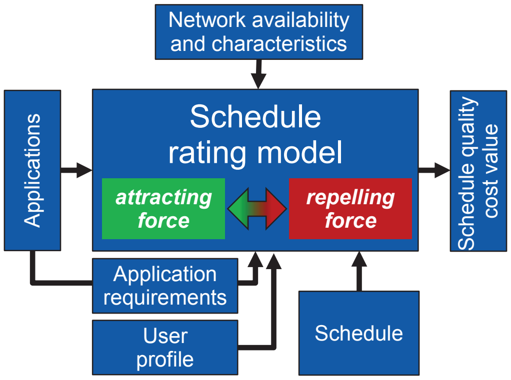

Figure 1: Schedule rating overview

network and time selection. It focuses on application QoS satisfaction with heterogeneous data traffic using heterogeneous networks.

To rate a schedule, we define a model of opposing forces. Firstly, we model an attracting force that pushes data to available networks. Secondly, we add a repelling force that pushes application data away from networks and time that violate application QoS requirements. Hence, a schedule has a certain tension of forces, for example from violation of a deadline or a minimum throughput requirement of a data flow. We express this tension in a cost function. The cost function serves as quality metric to assess schedules. A lower cost means that requirements of applications are satisfied better, and thus indicates higher schedule quality.

In a perfect schedule, data is pushed especially to QoS-matching networks. Moreover, delay tolerant data is moved to a point in time in which a matching network is available. Severe violations are avoided. As a result, the forces balance out transmissions of different applications according to user preference and to their integral QoS-defined priority.

A highlight of our schedule rating model is our novel throughput continuity model. To target the potential of time selection, it defines throughput requirements with tolerance to bursts and pauses. Bursty transfer may be more efficient in the dynamic scenario of mobile devices with fast changing network characteristics [5]. It enables data flow distribution to networks, which are available for a short time. As part of our novel schedule rating model, we therefore introduce an innovative throughput continuity model based on sliding windows.

However, the absolute cost function value provides no quantitative rating for schedule quality. For example, a scenario may have rare network capacity, which enforces violations. In this case, even the optimal schedule has a high cost function value. To make schedule quality assessable, we therefore provide a scoring system, which we call Normalized Rating Score (NRS). Instead of using absolute values, it states how much of the overall optimization potential of the scenario the scheduler exploits. This potential is defined by a relative value between two borders: the optimal score and the average score of a random scheduler as reasonable lower bound.

The schedule rating model needs information about its environment and about the data to transmit. Required information covers network availability and QoS characteristics, application QoS requirements and user preferences. A model overview is given in figure 1. Estimation or potential sources for this information are assumed to be available for offline analysis. We explicitly focus on the rating function, which assesses QoS and user preference satisfaction. This rating function enables detailed assessment of any scheduler and can be used to investigate weaknesses in model details and parameter choice for scored schedulers.

4. RATING MODEL COMPONENTS AND PARAMETER SPACE

Our schedule rating uses models for networks, data flows and user preferences. We summarize the defined parameters in table 1.

4.1 Network Model

Our network model covers throughput, latency, jitter and monetary costs. Network throughput is defined by bucket sizes in time slots. These bucket sizes $S_{t,n}$ represent the capacity that the network $n \in N$ can provide to the mobile node in time slot $t \in T$. Furthermore, latency L_n and jitter J_n are constant abstract parameters for each network. Additionally, network use leads to monetary costs $w_{mon,n}$ for the user. We model this monetary cost as constant abstract parameter for each network.

4.2 Data Flow Model

We define data flows by data amount, latency, jitter and throughput requirements and time limits. Hereby, a requirement for property x is defined in $\widehat{x}$, e.g., $\widehat{L}_f$ is a latency requirement for flow $f \in F$. The amount of data $\widehat{s}_f$ to be scheduled is abstracted using tokens that represent equally sized data chunks. During scheduling, tokens are allocated to time-slotted network buckets.

4.2.1 Throughput

We model throughput requirements as the number of tokens, which are allocated within a time window. We define a throughput limit using firstly a window size $\Delta\widehat{t}$. Its length is measured in number of time slots. Secondly, we define the amount of data inside the window in number of tokens $\widehat{\sigma}$. Modeling throughput using time windows and token amounts $\widehat{TP}_{min,f} = \{\Delta\widehat{t}_{min,f}, \widehat{\sigma}_{min,f}\}$ enables the definition of transmission continuity requirements. For example, a bufferable stream may be transmitted using large data bursts that fill a buffer fast, followed by potentially long pauses. We model it using a large window and many data tokens. In contrast, a live-stream requires continuous data transfer that we represent by a tiny window and proportionally less data tokens.

Transmission continuity follows from the integration of the new dimension of time into scheduling. It relaxes hard throughput requirements from alternative models that arose from imprecise static throughput models.

4.2.2 Start Time and Deadline

Adding the dimension of time into the scheduling model imposes timing requirements for known data flows. To model these timing requirements, we introduce an earliest start time $\widehat{t}_{st,f}$ and a deadline $\widehat{t}_{dl,f}$ for each data flow $f \in F$. They refer to the corresponding time slots.

4.2.3 Latency and Jitter

Latency requirements $\widehat{L}_f$ and jitter requirements $\widehat{J}_f$ are constants for each flow representing abstract upper limits. For rating, they are compared to the corresponding network characteristics.

4.2.4 Weights

Data flows of applications have individual requirements that we define through the previously introduced parameters. For each requirement, we define a linear weight that determines importance of the requirement for a flow. For example, latency requirement is important for a voice call, but not for a background download. The set of flow requirement weights is given in 1. Weights scale requirement violations of minimum throughput $w_{\widehat{TP}_{min,f}}$, start time $w_{\widehat{t}_{st,f}}$, deadline $w_{\widehat{t}_{dl,f}}$, latency $w_{\widehat{L}_f}$ and jitter $w_{\widehat{J}_f}$.

Table 1: Model parameters and variables

Description	Symbol
flow in flows to be scheduled	$f \in F$
time slot in overall planned time slots	$t \in T$
network in available networks	$n \in N$
network interface type i in interface types I	$i \in I$
number of tokens to be scheduled for flow f	$\widehat{s}_f$
capacity of network n in time slot t (network limit)	$S_{t,n}$
scheduled tokens of flow f in time slot t to network n	$s_{f,t,n}$
association $\in \{0,1\}$ of flow f in time slot t to network n	$a_{f,t,n}$
unscheduled tokens of flow f	u_f
number of interfaces of type i of the mobile node	k_i
size of max. throughput window of flow f in time slots	$\Delta\widehat{t}_{max,f}$
max. amount of tokens in max. throughput window of flow f	$\widehat{\sigma}_{max,f}$
size of min. throughput window of flow f in time slots	$\Delta\widehat{t}_{min,f}$
min. amount of tokens in min. throughput window of flow f	$\widehat{\sigma}_{min,f}$
violation of min. amount of tokens in min. throughput window for flow f in time slot t	$\sigma_{min,f,t}$
deadline requirement $\in T$ of flow f	$\widehat{t}_{dl,f}$
start time requirement $\in T$ of flow f	$\widehat{t}_{st,f}$
latency requirement of flow f	$\widehat{L}_f$
latency of network n	L_n
jitter requirement of flow f	$\widehat{J}_f$
jitter of network n	J_n
weight of parameter x for flow f	$w_{x,f}$
total cost function value	c
cost for model part y	c_y

This model leads to prioritization of flows with tough requirements. This behavior is intended. We explicitly do not implement a normalization. Thus, application QoS requirement strengths balance flow prioritization.

$$\widehat{W}_f = \{w_{\widehat{TP}_{min,f}}, w_{\widehat{t}_{st,f}}, w_{\widehat{t}_{dl,f}}, w_{\widehat{L}_f}, w_{\widehat{J}_f}, w_{user,f}\} \qquad (1)$$

4.3 User Preference Model

The user may prefer a specific application or flow. Therefore, we introduce another linear weight $w_{user,f}$ that influences prioritization of flows. This enables personalization of application priorities.

To give users a trade-off trigger between network performance and monetary transmission cost, we introduce the balancing parameter willingness-to-pay w_{wtp}. We define willingness-to-pay inversely. A low value results in a high willingness of the user to pay for more network performance. It trades off accumulated flow violations to monetary cost.

4.4 Discussion

Our network and data flow models both integrate the dimension of time. This enables accurate rating of transmission volumes and flow shapes. In particular, the novel throughput model enables flow continuity optimization, using flow rate adaption, continuous data transfer or allowing bursts. Weights provide flexible requirement definition, which enable modeling of heterogeneous data traffic with time-dependent transmission constraints.

5. BEHAVIORAL MODEL DEFINITION

In the following, we introduce the equations that define the cost function for rating and a set of schedule feasibility constraints. Within these equations, we use the identity function $\mathbb{I}(x = y)$, which is 1 if x is equal to y, else 0. Furthermore, we use signum function $sgn(x)$, which is 1 for positive x, 0 if $x = 0$ and -1 for negative x. Furthermore, all parameters and decision variables are of type integer. In addition, $a_{f,t,n}$ is limited to $[0, 1]$.

The core of our model is the cost function. We define it according to an analogy to opposing forces. The attracting forces foster token allocation to networks. They cover a punishment for unscheduled tokens and violation of minimum throughput application requirement. In contrast, the repelling forces punish tokens that are allocated at networks or time slots, which violate application requirements. They cover firstly violation models for latency, jitter, start time and deadline and secondly monetary cost model.

5.1 Token Allocation

This model regulates how tokens can be allocated to networks and time slots.

5.1.1 Token Scheduling

A token can either be scheduled $s_{f,t,n}$ or unscheduled u_f. In sum, the number of tokens of a flow is limited by $\widehat{s}_f$. We model constraint this `C1` in equation 4. Scheduled tokens $s_{f,t,n}$ lead to data transmission, which implies monetary cost c_{mon} according to equation 5. This monetary cost applies a repelling force, which pushes tokens away from expensive networks. The force is scaled by willingness-to-pay w_{wtp}. In contrast, unscheduled tokens u_f lead to violation $c_{u,f}$ as shown in equation 6. They take effect in an attracting force, which pulls tokens to networks. Furthermore, the sum of scheduled flow tokens $s_{f,t,n}$ must not exceed available resources of networks, modeled by the bucketsize $S_{t,n}$. Constraint `C2` in equation 7 ensures this limit.

5.1.2 Available Interfaces

In our model, the mobile node has a specified number of network interfaces k_i of the type $i \in I$, e.g. WiFi or cellular network. The mobile node cannot connect to more networks of the same type i at the same time, than dedicated interfaces available. We model this constraint `C3` according to equation 8.

5.1.3 Flow Migration Model

Flow migration, or handover from one network to another, leads to protocol overhead and might lead to performance degradation during the process. It contributes to the repelling forces, pushing tokens away from all networks except from the currently used. To suppress frequent and unnecessary flow migration, often referred to as ping-pong effects, each migration process contributes linearly to the cost function. Flow migration is complex to detect from our scheduling matrix, because scheduled flows can pause for a while.

5.1.4 Network Association Model

To detect flow migrations, we associate each flow at each time slot to a single network that is used for potential transmission $a_{f,t,n}$. We ensure this by equation 9. To provide coincidence between network association $a_{f,t,n}$ and scheduled tokens $s_{f,t,n}$, we introduce constraint 10. It ensures that token assignment presumes flow association to the corresponding network. In contrast, an association does not require an assignment of tokens. Therefore, a network association for each flow exists in each time slot. This enables a simple way to identify flow migration: checking of association changes in consecutive time slots. We calculate the cost from flow migration in equation 11. It counts the number of flow migrations and weights it linearly by w_{mig} to calculate flow migration cost c_{mig}.

5.2 Throughput Model

Our throughput model defines window sizes and limits for the number of tokens inside of them. The model assesses limits for each time step, sliding the window along the planned time span. This is shown in figure 2. Many data flows require a minimum throughput. We add an attractive force to the model that pushes data tokens to networks to affirm minimum flow throughput requirement. We model this force in equation 13. It counts the number

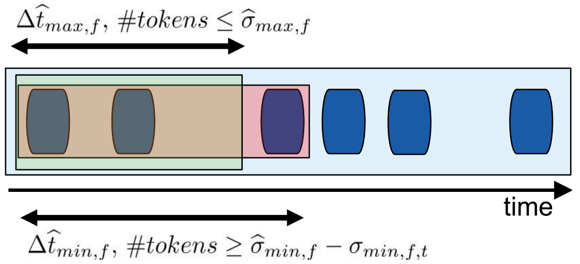

Figure 2: Throughput model: window size is defined by $\Delta\widehat{t}$, number of required tokens by $\widehat{\sigma}$. $\#tokens$ stands for the number of scheduled tokens of a flow in the time window over all networks.

of scheduled tokens $s_{f,t,n}$ inside each window over all networks. The sum of tokens in the window should at least be equal to the required number $\widehat{\sigma}_{min,f}$. Violation is allowed by the additional term of $\sigma_{min,f,t}$, which contributes to the cost function and is therefore minimized.

Moreover, the model also requires an upper throughput model, as shown in equation 12. This constraint `C4` ensures, that a scheduler does not plan a higher throughput than the source can deliver. An example is a live-stream. Data cannot be transmitted earlier with high throughput bursts because it has not been generated yet. It is a hard constraint, which cannot be violated. Therefore, it does not contribute to the cost function.

5.2.1 *Throughput Violation Normalization*

To derive the cost function impact for minimum throughput violation $c_{\widehat{TP}_{min,f}}$, we define a cost normalization for violation token count $\sigma_{min,f,t}$ in equation 14. To treat flows equally from their data amount, we normalize the violation cost to the number of tokens of a flow. Thus, we add the linear parameter $\widehat{\sigma}_{min,f}$ to the cost function calculation. Additionally, a token gap, which causes violation, is counted multiple times, while a window slides over it. A long window covers the gap at more sliding steps and leads to higher violation token count. To avoid violation strength dependence on window size $\Delta\widehat{t}_{min,f}$, we normalize cost to it. To counteract contradiction with start time and deadline requirements, the minimum throughput requirement is only active within the time slots in between. The latter normalization underrates violation near start time and deadline, because gaps are covered fewer times by the sliding window. For example, the time slot at start time is covered once only: in the first sliding step of the window. Accordingly, the model penalizes minimum throughput violation at the start and the end lower than in the middle. Indeed, this approximates reality. It provides two advantages: Firstly, real flow control mechanisms increase throughput gradually. The model error endorses lower throughput at flow start that represents real transmission behavior. Secondly, decreasing data rate towards the deadline implies that data should be transmitted earlier. This favors conservative data buffer filling towards the deadline, which helps to finish data transmission in time.

5.3 Start Time, Deadline, Latency and Jitter

Start time, deadline, latency and jitter violations create repelling forces that push tokens away from non-matching networks. Violation contributes to the cost function with quadratic impact, as shown in equations 15 - 18. For example, a severe deadline violation causes a higher force than many slight violations. As a result, the model avoids severe violations. Note that the quadratic impact factors do not depend on the schedule, but can be pre-calculated. Therefore, the model is still linear. Corresponding weights and the number of violating tokens scale violations linearly.

5.4 Cost Function

The absolute cost c reaches its minimum at an equilibrium of forces with an optimal schedule. We define absolute cost in equation 2. It covers firstly monetary cost, secondly flow migration cost and thirdly the flow requirement violation cost from equation 3 with their user-defined weights.

Finally, the objective of the optimization is to minimize c, subject to equations 3..18.

$$c = c_{mon} + c_{mig} + \sum_{f \in F} (w_{user,f} \cdot c_f) \tag{2}$$

$$c_f = c_{u,f} + c_{\widehat{TP}_{min,f}} + c_{dl,f} + c_{st,f} + c_{L,f} + c_{J,f} \tag{3}$$

Token allocation model:

$$\texttt{C1:}\ \forall f \in F: \sum_{\substack{t \in T,\\ n \in N}} s_{f,t,n} + u_f = \widehat{s}_f \tag{4}$$

$$c_{mon} = w_{wtp} \cdot \sum_{n \in N} \left(w_{mon,n} \cdot \sum_{\substack{f \in F,\\ t \in T}} s_{f,t,n} \right) \tag{5}$$

$$c_{u,f} = w_{u,f} \cdot u_f \cdot w_{user,f} \tag{6}$$

$$\texttt{C2:}\ \forall t \in T, n \in N: \sum_{f \in F} s_{f,t,n} \leq S_{t,n} \tag{7}$$

$$\begin{gathered}\texttt{C3:}\ \forall t \in T, i \in I: \\ k_i \geq \sum_{n \in N} \left(\mathbb{I}(i_n = i) \cdot \sum_{f \in F} sgn(s_{f,t,n}) \right)\end{gathered} \tag{8}$$

Network association and flow migration model:

$$\forall f \in F, t \in T: \sum_{n \in N} a_{f,t,n} = 1 \tag{9}$$

$$\forall f \in F, t \in T, n \in N: \ a_{f,t,n} \geq sgn(s_{f,t,n}) \tag{10}$$

$$c_{mig} = w_{mig} \cdot \sum_{\substack{f \in F,\\ n \in N}} \sum_{\substack{t \in T|\\ t \geq 2}} (1 - \mathbb{I}(a_{f,t-1,n} = a_{f,t,n})) \tag{11}$$

Throughput model:

$$\texttt{C4:}\ \forall f \in F, t_0 \in T: \widehat{\sigma}_{max,f} \geq \sum_{t=t_0}^{t_0+\Delta\widehat{t}_{max,f}} \sum_{n \in N} s_{f,t,n} \tag{12}$$

$$\begin{gathered}\forall f \in F, t_0 \in T: \\ \widehat{\sigma}_{min,f} \leq \sum_{t=t_0}^{t_0+\Delta\widehat{t}_{min,f}} \sum_{n \in N} s_{f,t,n} + \sigma_{min,f,t}\end{gathered} \tag{13}$$

$$\forall f \in F: c_{\widehat{TP}_{min,f}} = \frac{w_{\widehat{TP}_{min,f}}}{\Delta\widehat{t}_{min,f} \cdot \widehat{\sigma}_{min,f}} \cdot \sum_{t=\widehat{t}_{st,f}}^{\widehat{t}_{dl,f}-\Delta\widehat{t}_{min,f}} \sigma_{min,f,t} \tag{14}$$

Deadline, start time, latency and jitter models:

$$\forall f \in F: \ c_{dl,f} = w_{\widehat{t}_{dl,f}} \cdot \sum_{\substack{t \in T, \\ n \in N}} sgn(s_{f,t,n}) \cdot \max(0, t - \widehat{t}_{dl,f})^2 \tag{15}$$

$$\forall f \in F: \ c_{st,f} = w_{\widehat{t}_{st,f}} \cdot \sum_{\substack{t \in T, \\ n \in N}} sgn(s_{f,t,n}) \cdot \max(0, \widehat{t}_{st,f} - t)^2 \tag{16}$$

$$\forall f \in F: \ c_{L,f} = w_{\widehat{L}_f} \cdot \sum_{\substack{t \in T, \\ n \in N}} sgn(s_{f,t,n}) \cdot \max(0, L_n - \widehat{L}_f)^2 \tag{17}$$

$$\forall f \in F: \ c_{J,f} = w_{\widehat{J}_f} \cdot \sum_{\substack{t \in T, \\ n \in N}} sgn(s_{f,t,n}) \cdot \max(0, J_n - \widehat{J}_f)^2 \tag{18}$$

6. COMPLEXITY ANALYSIS

Problem definition: *Given a schedule* S*, a set of networks and a set of data flows, is* S *a feasible schedule and is* S *of minimal cost?*

The presented model defines a polynomial-time cost function in equation 2 and sub-functions and additional polynomial-time constraints `C1-C4`. Hence, the feasibility and the cost of a schedule s can be determined within polynomial time. However, it is not possible to verify in polynomial time if s is of minimal cost, given $P \neq NP$.

To analyze the problem complexity, we start with a separation. The effect of the cost function parameters L, J, $w_{mon,n}$ and $w_{u,f}$ follows from two factors. Firstly from the match between network and flow and secondly from the number of scheduled tokens. Time does not influence those equations. Thus, a time-independent solution subspace of the scheduling problem can be derived, when we consider only $s_{f,n}$, as shown in equation 19.

$$s_{f,n} = \sum_{t \in T} s_{f,t,n} \tag{19}$$

$$\begin{aligned}\forall f \in F, n \in N \ : v_{f,n} = sgn(s_{f,n}) \cdot \Big(w_{\widehat{L}_f} \cdot \max(0, L_n - \widehat{L}_f)^2 \\ + w_{\widehat{J}_f} \cdot \max(0, J_n - \widehat{J}_f)^2 \Big) + s_{f,n} \cdot (w_{mon,n} - w_{u,f})\end{aligned} \tag{20}$$

We are able to derive a time-constant value $v_{f,n}$ from the separated properties, as shown in equation 20. This separated part of the optimization is equivalent to multiple-demand bounded multiple knapsack problem. Flows with their individual network-matching $v_{f,n}$ represent multiple, different demands. Furthermore, the number of tokens of a flow is equivalent to the bounded number of equal items. Limited knapsack sizes are expressed by limited network resources, whereby multiple networks with individual capacity exist. We argue that the defined scheduling problem is NP-hard, because knapsack is a sub-problem that is NP-complete and the scheduling is not in NP.

The presented scheduling problem contains an additional dimension that makes it harder to solve: time. However, the omitted, time-dependent parts of the cost function may dominate the cost value of a knapsack solution from the split-off sub-problem. Therefore, the time-dependent parts cannot be considered separately. These parts are firstly our throughput violation model, secondly the start time and deadline violation models and finally the flow migration cost model. For scheduling, we therefore cannot apply the well-known heuristics of knapsack problem.

7. HEURISTIC SCHEDULERS

To fill the gap of heuristic schedulers, we present three approaches and a random baseline. To make the three scheduler approaches comparable, we apply the same heuristics for flow prioritization and network matching to all of them. Therefore, they only differ in time selection in their search strategy. This decision follows the goal of this work, to investigate the effect of the paradigm change of integrating time selection as explicit part into network selection. In the following, we firstly present the applied heuristics and constraint checkers and secondly explain the search strategies of the schedulers.

7.1 Heuristic Design

Substantial elements in the scheduling process are flow prioritization and network matching. We present heuristics for these two elements.

7.1.1 Flow Prioritization

For flow prioritization, we rely on the individual flow requirements. According to the well-known Most-Constrained-First theorem in search heuristics for Constrained Satisfaction Problems (CSP), we schedule flows first, which are expected to be most restricted ones. Therefore, we create a function that evaluates the restrictiveness $\widetilde{r}_f$ of a flow f. In complexity analysis in section 6, we already identified the time-invariant cost function terms. These are latency, jitter, monetary cost and number of unscheduled chunks of a flow. We use these cost function terms linearly in our restrictiveness heuristic. In addition, we linearize minimum throughput requirement by the average throughput within the corresponding time window. Our restrictiveness heuristic sums up all restrictive flow requirements, multiplied by their corresponding weights, as shown in equation 21.

$$\begin{aligned}\widetilde{r}_f = w_{user,f} \cdot \Big[w_{u,f} \cdot w_{user,f} + w_{\widehat{J}_f} \cdot \widehat{J}_f^2 + w_{\widehat{L}_f} \cdot \widehat{L}_f^2 \\ + w_{\widehat{TP}_{min,f}} \cdot \widehat{TP}_{min,f,average} \Big]\end{aligned} \tag{21}$$

7.1.2 Network Matching

For network matching, we design a cost heuristic, which is based on the cost function of the proposed rating model and approximates time-dependent terms. It covers the same requirements as our restrictiveness heuristic but compares them to the characteristics of a network to estimate their match. In addition, it covers the monetary cost induced from network use. We bisect the terms into attracting and repelling forces and give the attracting forces a negative influence on the result. The derived cost heuristic is shown in equation 22 and gives an estimation of the cost $\widetilde{c}_{f,n}$ of a token assignment of a flow f to network n. Note, that this estimated cost is usually negative, since token assignment is supposed to improve the schedule.

$$\begin{aligned}\widetilde{c}_{f,n} = w_{user,f} \cdot [\ w_{\widehat{J}_f} \cdot \max(0, J_n - \widehat{J}_f)^2 \\ + w_{\widehat{L}_f} \cdot \max(0, L_n - \widehat{L}_f)^2 - w_{u,f} \cdot w_{user,f} \\ - w_{\widehat{TP}_{min,f}} \cdot \frac{TP_{n,average} - \widehat{TP}_{min,f,average}}{\Delta \widehat{t}_{min,f} \cdot \widehat{\sigma}_{min,f}}\ ] \\ + w_{mon,n} \cdot w_{wtp}\end{aligned} \tag{22}$$

7.2 Constraint Satisfaction

All presented schedulers rely on search algorithms for constrained optimization problems. Constraints `C1`-`C4` invalidate a large share of the solution space. To stay within the valid solution space during search, we apply a forward checking strategy for all four schedulers. The forward checking disqualifies token assignments according to the constraints of the optimization. `C1`, `C2` and `C3` limit the token amount for an assignment. For each assignment, we apply the minimum operator to remaining tokens of the flow, remaining capacity of the network in time slot and averaged maximum throughput of the flow. C4 constraints network selection based on available network interfaces of the mobile node. For each network selection, it validates if the network is already planned to be used or if for this time slot a network interface of matching type is free. In addition, we simplify start time and deadline violation model to constraints. No data is scheduled outside these limits. In applying forward checking, our search heuristics do not require a backtracking strategy for constraint satisfaction.

7.3 Schedulers

The schedulers apply the defined heuristics and differ in their strategy of time selection. We follow this approach to investigate the impact of time in search strategies on schedule quality and thereby target the main goal of this work.

7.3.1 Network Selection (NS)

The Network Selection scheduler splits the search space by time slot and treats each slot separately. Therefore, time has no impact on its search strategy. This follows the usual state-of-the-art approach of network selection schemes. It starts with sorting of flows to assess which flows should be prioritized during scheduling, using restrictiveness heuristic $\widetilde{r}_f$ according to equation 21.

For each flow, it sorts available networks according to their match determined by cost heuristic $\widetilde{c}_{f,n}$ according to equation 22 and tries to assign as many tokens of the flow as possible to the networks in the sorted list, starting with the best match. For token assignment, we apply forward checking to ensure that all constraints are satisfied and the resulting schedule is valid.

7.3.2 Opportunistic Network Selection (ONS)

Opportunistic Network Selection is based on the NS. In addition, it adds an opportunistic term to token assignment, which enables decision to refuse scheduling to bad matching networks in current time slot. It therefore introduces a new parameter, the cost limit c_{lim}. The algorithm only adds networks to the sorted network list, whose cost heuristic result is lower than the limit $\widetilde{c}_{f,n} < c_{lim}$. This change in strategy adds a statistical time impact because some data, which could be scheduled in current time slot, is delayed for potential later transmission. Time selection of ONS follows state-of-the-art resource allocation approaches. In addition, we combine their strategies with network selection.

7.3.3 Transmission Planning (TP)

Transmission Planning assumes to have further knowledge on future available networks. Instead of splitting the search space into independently scheduled time slots like NS and ONS, it first focuses on network matching.

TP starts with a flow prioritization. Like NS and ONS, TP applies $\widetilde{r}_f$ to sort flows according to their restrictiveness. Secondly, for each data flow, networks are sorted according to their match in network characteristics using the cost heuristic $\widetilde{c}_{f,n}$. As last step before token assignment, it selects transmission time. It therefore checks the overlap of network availability and the time window between flow start time and deadline. This novel strategy focuses on QoS satisfaction. Instead of ignoring optimization potential of time completely like NS, or covering it statistically like ONS, TP explicitly employs time selection as optimization step.

7.3.4 Random Scheduler

The random scheduler follows the design of NS. Instead of applying heuristics for flow and network sorting, it uses a shuffle function. During token assignment, forward checking still ensures validity of the resulting schedules.

8. EVALUATION

In this evaluation, we give evidence that integrating the time domain into network selection is beneficial. As second contribution, we show how to apply our novel rating scheme to identify strengths and weaknesses of schedulers. In the following, we explain our evaluation metrics and setup and discuss the results and improvement potentials of the three heuristic schedulers.

8.1 Evaluation Metrics

Our rating model defines an appropriate quality measure in its cost function. It creates a comparable rating measure for any schedule. However, the absolute cost function values alone provide no evidence on quality of a schedule. Depending on the scenario, a schedule may have violations and therefore a high cost function value, even though it is ideal. We therefore compare scheduling cost function values to bounding values for the given scenario.

An optimization determines a schedule of minimum cost for the scenario, but has a long execution time. This is not convenient for online execution. Nevertheless, it provides the upper quality bound and is therefore a perfect candidate to analyze improvement potentials of schedulers. To assess schedule quality, we additionally require a reasonable lower quality bound. We select a random scheduler for this purpose. It provides feasible, but low quality schedules. Every reasonable scheduler should return results better than average random. The two bounds enable relative quality assessment.

We introduce the Normalized Rating Score (NRS) in equation 23. It provides a score from 0 to 1, which informs about the quality of the schedule s. 0 means, that it is as bad as average random schedules. In contrast, a score of 1 means that the scheduler reached optimal performance. From NRS, the overall optimization potential of the scheduler can be estimated. To clarify which details of the scored scheduler lead to this waste of potential, we introduce Relative Detail Score (RDS) in equation 24. It provides cost information on the rated requirement satisfaction criteria of the cost function. RDS determines the deviation of criterion v from this of the optimal schedule, normalized to the absolute cost function optimization potential of schedule s. A value of 0.2 means, that criterion v (e.g. minimum throughput requirement) contributes 20% to the wasted optimization potential. In contrast, a value of -0.1 means, that the scheduler performs 10% better than the optimum in criteria v. This gives hint, that the scheduler should be less restrictive on criterion v in order to provide space for optimization of other criteria. In these two cases, a parameter change or model refinement on criterion v should be considered. A value of 0 means, that the scheduler brings criteria v perfectly in balance to reach optimal schedules.

Additionally, computational complexity of schedulers can lead to long execution times, which could make them infeasible for real world use. Therefore, we use execution time as last evaluation metric to assess schedulers.

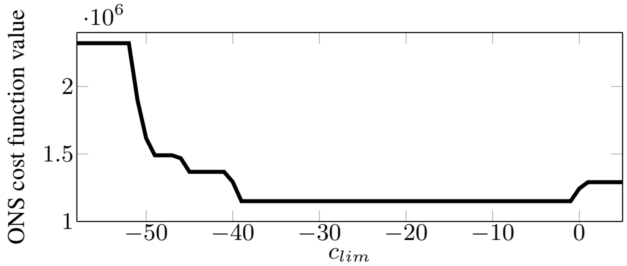

Figure 3: ONS cost function value over c_{lim}

$$NRS(s) = \frac{c_{Rnd} - c_s}{c_{Rnd} - c_{Opt}}; \tag{23}$$

$$RDS_v(s) = \frac{c_{v,s} - c_{v,Opt}}{c_s - c_{Opt}}; \tag{24}$$

The three evaluation metrics of NRS, RDS and execution time provide means to analyze firstly the overall scheduler performance, secondly weaknesses and optimization potential of schedulers and thirdly real-time applicability of schedulers.

8.2 Evaluation Setup

For evaluation, we generate randomized scenarios with 16 flows, eight networks and vary the scenario size by the number of time slots from 25 to 400. The simulated networks cover two cellular networks, which are always available and six WiFis, which are available only for certain time windows. In comparison to WiFi, simulated cellular networks provide higher latency, equal to lower jitter, lower to equal throughput and cause higher cost. Data flows are composed of four typical traffic classes: Live-stream, bufferable stream, interactive and background with randomized requirement values within typical ranges and traffic share [10, 3]. Mobile nodes furthermore have one cellular network interface and one WiFi interface. For each scenario size, we generate 30 randomized scenarios and measure quality and execution time. For random scheduler, we execute 100 runs for each scenario. The results show about normal distribution (Anderson-Darling test: execution time p = 0.72, cost p = 0.78). We can therefore use average values of the 100 runs for NRS. For optimization, we use IBM CPLEX Branch&Cut solver. We implemented the random and heuristic schedulers in Java. For each simulation instance, we use one core of a server machine with Intel Xeon E5-2643 v3 @ 3.4GHz and 512 GB RAM.

8.3 ONS Opportunistic Parameter Tuning

The ONS scheduler delays token assignments when expected cost $\tilde{c}_{f,n}$ is higher than c_{lim}. Therefore, schedule quality of ONS depends on this parameter. The lower its value is, the higher is the probability to delay traffic. A too low value will always effect token assignment skipping. Figure 3 shows the cost function value depending on c_{lim}. We observe a relatively large window for optimal choice of the parameter. For evaluation, we select $c_{lim} = -10$.

8.4 Evaluation Results and Discussion

We first present absolute cost function values and execution time to give an impression on the data set and then rate the different schedulers using our novel rating scheme, consisting of the cost function and the metrics NRS and RDS. We show the Optimal Scheduler (Opt), the classical Network Selection (NS), the Opportunistic Network Selection (ONS), the Traffic Planning (TP) and the Random Scheduler (Rnd). For all plots, we vary scenario size in time slots on the x axis.

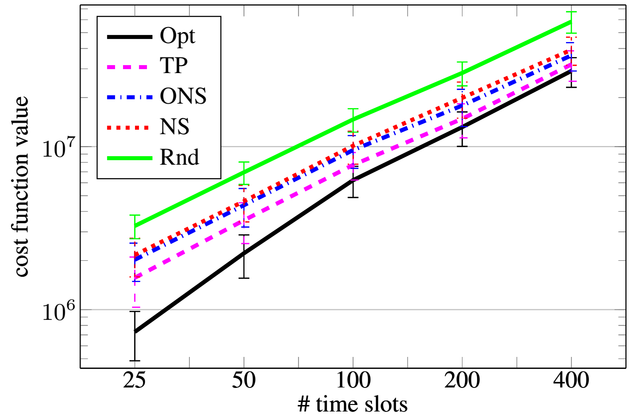

Figure 4: Cost function results of schedulers over scenario size

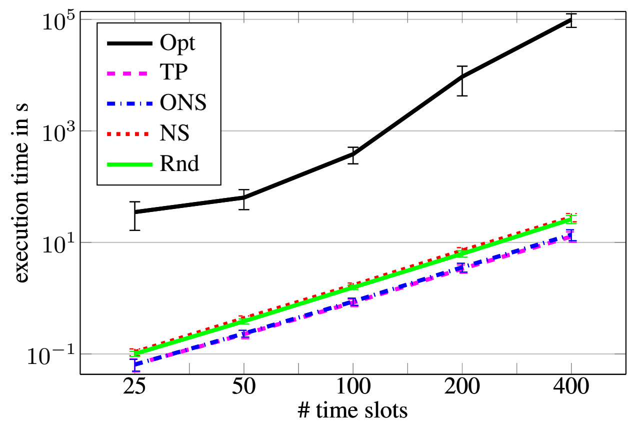

Figure 5: Execution times of schedulers over scenario size

8.4.1 *Absolute Cost Value and Execution Time*

The absolute cost in figure 4 shows an overall increase of values towards larger scenarios for all schedulers. For cost function value, schedulers keep in a strict order in every scenario. All heuristic schedulers show approximately a linear increase of cost function value with scenario size while the optimization scheduler result shows a concave curve, slowly approximating to the other schedulers. This admits the claim that our heuristic results converge towards the optimum with increasing scenario size.

For execution time, we observe exponential increase for the optimization, resulting in an average execution time of 98766s (27.4h) for solving the largest simulated scenario size. In contrast, all heuristic schedulers show an average execution time of less than 29s for the same scenario size. Note, that the two online schedulers NS and ONS can distribute their workload over the real scenario duration. All three of them provide real-time compliant solutions for scheduling.

8.4.2 *Rating Score Analysis*

We propose to use our metric NRS in addition to absolute values and RDS for detail analysis. NRS indicates how much of the overall optimization potential the scheduler uses. RDS shows how much the model sub-function violations diverge from optimal cost distribution. This provides inference about which parameter choices or models of a scheduler contain the main weaknesses. We use these two metrics to analyze the three proposed heuristic schedulers.

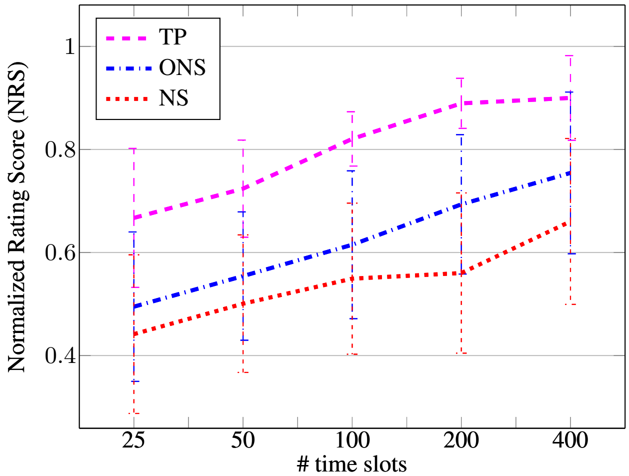

Figure 6: Normalized Rating Score (NRS) over scenario size

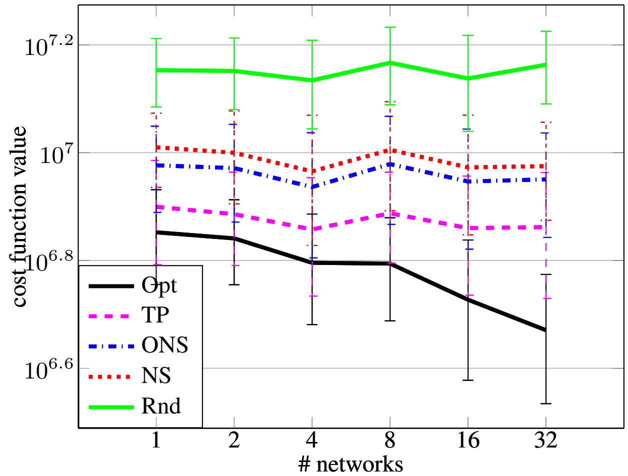

Figure 7: Absolute cost function value over number of networks, scenario size = 100 time slots

Normalized Rating Score (NRS) Analysis.

In figure 6, we show the NRS of the three heuristic schedulers NS, ONS and TP. Like in absolute cost value comparison, we observe a strict order in performance of the three schedulers and a convergence towards the optimum with rising scenario size.

In contrast to figure 4, the used optimization potential of each scheduler gets obvious. TP uses in average 18% more of this potential than ONS and 26% more than NS. Note, that improvement of NRS gets harder with rising absolute NRS score. Since all three schedulers only differ on the impact of time in search strategy, this observation confirms our initial hypothesis: For the given scenario with non-constant network characteristics, there is a substantial impact of time in network selection.

But why do the schedulers' NRS converge towards the optimum with rising scenario sizes? The answer originates from the fixed number of networks in the scenario. With rising scenario duration, the relative number of network-changes decreases linearly. If network-changes are rare, the overall optimization potential of joint network and time selection decreases and the simplified throughput heuristics converge towards the optimum. To show this effect, we vary the number of networks and, hence, adapt the frequency of network-changes over time. This is illustrated in figure 7. The one-network case depicts the limited optimization potential of state-of-the-art resource allocation. Moreover, we see how the optimization potential, i.e. the difference between the optimal and averaged random-scheduler cost function values, increases with the number of networks. This affirms our initial hypothesis, that joint network and time selection for mobile nodes leads to synergies which provide unknown optimization potentials. However, the performance of the three heuristic schedulers stay nearly identical. This leads to a relative decrease of performance in NRS score towards scenarios with high optimization potential. The heuristic schedulers' behavior clarifies the challenge in development of new schedulers, which use the new identified optimization potential in a better way.

Relative Detail Score (RDS) Analysis.

To identify the origins of the weaknesses of the schedulers and to find out how to improve them, we analyze the RDS for the three proposed schedulers, shown in figure 8. Note that RDS is normed on individual scheduler performance and should therefore not be used for comparison of schedulers to each other. Keep in mind that the heuristics are kept simple by intention to be able to apply them equally to all three schedulers in order to investigate the impact of time in network selection. The development of more elaborated schedulers exceeds the scope of this paper.

We can observe a good balance of the heuristics for the latency and jitter model. Their performance turn out as expected since we reuse the original rating model for the heuristic.

Unsurprisingly, time limits model, namely start time and deadline violations, overperform slightly because the heuristic schedulers forbid violation of start times or deadlines. However, any overperformance can have essential impact on other categories. For example, the heuristic schedulers always hold even soft deadlines. Nevertheless, their violation provides high potentials to assign additional tokens to well-matching networks. To cope with this weakness, a trade-off heuristic for time limit violation is required.

Unscheduled tokens and throughput violation play a major role in wasted optimization potential. For the two network selection strategies NS and ONS, unscheduled token violation impact decreases with rising scenario size. This is the case because the probability for existing later time slots with free capacity in matching networks rises with higher overall scenario duration.

In addition, the heuristics select networks based on restrictiveness values of single flows only. Therefore, especially live-streams with high restrictiveness dominate the network selection and network interfaces get occupied early in process from this strategy. This behavior seems to underrate the selection of high throughput networks and causes throughput shortage. As observed, this leads to many unscheduled tokens especially from high bandwidth flows with low restrictiveness like bufferable streaming applications and background traffic. A better approach could use a collaborative throughput demand assessment to select networks based on peek demands. However, this collaborative assessment also requires new decision models for arising questions including: Which flows should share a medium in a time slot?

For ONS and TP, monetary cost contributes to the strategy of opportunistic respectively explicit time selection. We suppose that its overperformance for the two schedulers correlates with the number of unscheduled chunks. Since less chunks are scheduled from both strategies, the overall transmission cost drops as well. In NS, this effect is dominated by a much worse network selection due to the lack of time selection. The rising impact over time can be explained by the nature of the scheduler. NS assigns tokens to the best matching network available as early as possible. If suiting, cheaper networks are available at a later point in time, the optimization potential rises but stays unused by NS. Therefore, monetary cost impact rises with increasing scenario size for NS.

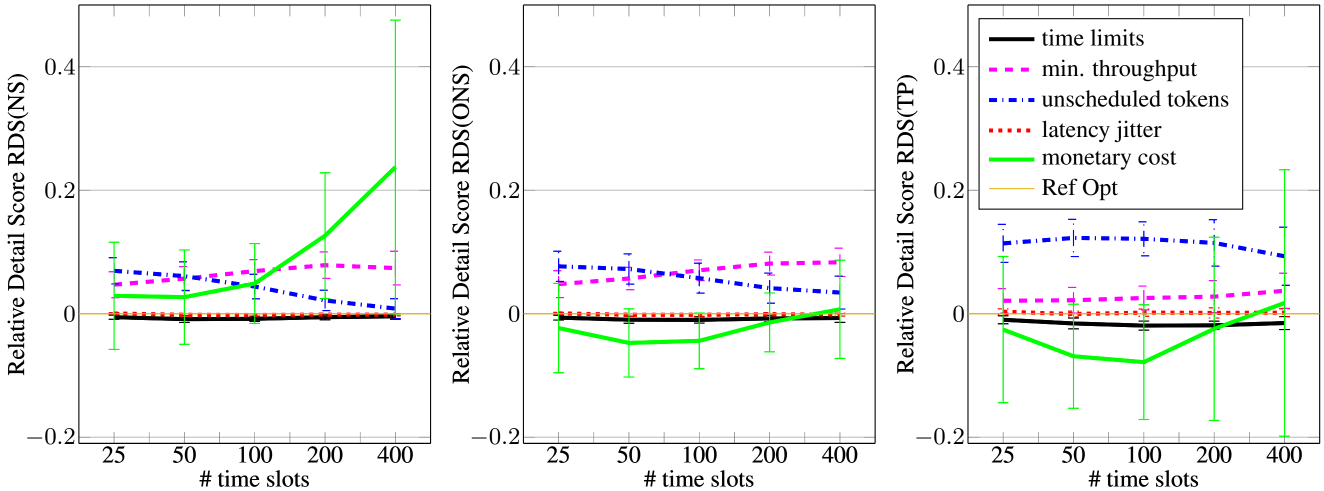

Figure 8: Relative Detail Scores (RDS) over scenario size for NS (left), ONS (center), TP (right)

9. CONCLUSION

We hypothesized, that in rapidly changing network environments, time selection within network selection has a notable impact on user-perceived network performance. To the best of our knowledge, time selection and network selection have only been considered separately so far. To investigate our hypothesis, we define a rating scheme able to assess joint network and time selection. Our novel schedule rating scheme assesses application QoS-requirement satisfaction and provides an innovative throughput model, which integrates the impact of bursty data transfer. This novel rating scheme is flexible enough to model extensive application QoS requirements, network characteristics and user preferences. It acts as a tool to investigate our initial hypothesis and is the first contribution of this work.

To approach our second contribution, the time-impact analysis on network selection for mobile nodes, we present three heuristic schedulers, which differ in their strategy of time selection. Two of them follow state-of-the-art approaches with classical and opportunistic network selection. The third explicitly integrates time selection into network selection. We use our novel rating model to investigate the impact of time on network selection on these three schedulers. We furthermore present how to apply the new tools to analyze weaknesses of schedulers. As result, we observe a strong impact of 26% respectively 18% of improvement when using the joint selection instead of the classical respectively an opportunistic approach. This confirms our initial hypothesis. This new insight reveals a path to exploit new potentials of network usage optimization for mobile nodes: a paradigm change from classical network selection strategies towards joint network and time selection.

10. REFERENCES

[1] F. Bari and V. C. M. Leung. Automated Network Selection in a Heterogeneous Wireless Network Environment. *IEEE Network*, 21(1), 2007.

[2] N. Bui and J. Widmer. Mobile Network Resource Optimization under Imperfect Prediction. *IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.

[3] Ericsson. Mobility Report. Technical report, 2016.

[4] K. Johnsson and D. Cox. An adaptive cross-layer scheduler for improved QoS support of multiclass data services on wireless systems. *IEEE Journal on Selected Areas in Communications*, 23(2), 2005.

[5] Z. Lu and G. D. Veciana. Optimizing Stored Video Delivery For Mobile Networks : The Value of Knowing the Future. *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2013.

[6] I. Malanchini, M. Cesana, and N. Gatti. Network Selection and Resource Allocation Games for Wireless Access Networks. *IEEE Transactions on Mobile Computing*, 12(12), 2013.

[7] K. R. Rao, Z. S. Bojkovic, and B. M. Bakmaz. Network Selection in Heterogeneous Environment : A Step toward Always Best Connected and Served. *Proceedings of International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, 2013.

[8] T. Rueckelt, D. Burgstahler, F. Englert, C. Gottron, and S. Zöller. A Concept for Vehicle Internet Connectivity for Non-Safety Applications. In *Proceedings of the 39th IEEE Local Computer Networks Conference (LCN)*, 2014.

[9] T. Rueckelt, F. Jomrich, D. Burgstahler, D. Böhnstedt, and R. Steinmetz. Publish-Subscribe-Based Control Mechanism for Scheduling Integration in Mobile IPv6. In *Proceedings of the 40th Local Computer Networks Conference (LCN)*, number October, 2015.

[10] Sandvine. Global Internet Phenomena Asia-Pacific & Europe. Technical report, 2015.

[11] R. Trestian, O. Ormond, and G.-M. Muntean. Game Theory-Based Network Selection: Solutions and Challenges. *IEEE Communications Surveys & Tutorials*, 14(4), 2012.

[12] L. Wang and G.-S. G. Kuo. Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks - A Tutorial. *IEEE Communications Surveys & Tutorials*, 15(1), 2013.

[13] Q. Wu, S. Member, Z. Du, S. Member, and P. Yang. Traffic-Aware Online Network Selection in Heterogeneous Wireless Networks. *IEEE Transactions on Vehicular Technology*, 65(1), 2016.

[14] K.-k. Yap, T.-Y. Huang, Y. Yiakoumis, S. Chinchali, N. McKeown, and S. Katti. Scheduling packets over multiple interfaces while respecting user preferences. *Proceedings of the 9th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2013.