Layered Cooperative Resource Sharing at a Wireless SDN Backhaul

Lorenzo Ferrari, Nurullah Karakoç, Anna Scaglione, Martin Reisslein
School of Electrical, Computer and Energy Engineering
Arizona State University, Tempe, AZ
lferrari, nkarakoc, ascaglio, reisslein@asu.edu

Akhilesh Thyagaturu
Intel Cellular & Devices Group (iCDG)
Intel Corporation, San Diego, CA
akhilesh.s.thyagaturu@intel.com

***Abstract*—This paper describes a unifying optimization framework to share backhaul network resources across different operators and wireless platforms. The architecture we consider, named LayBack, requires introducing a unifying Software Defined Network (SDN) orchestrator, sited where their respective traffic streams meet: at the wireless network backhaul. The work we present proposes a scalable decomposition of the resource allocation problem across different layers and time-scales.**

I. Introduction

Today there is only very limited statistical multiplexing (sharing) of spectrum resources across wireless operators and technologies [1], [2]. There have been efforts in wireless standards [3], [4] and in academic research [5] to define architectures aimed at changing the status quo, e.g., enabling resource sharing only among individual LTE cells [6], [7] but they generally have had limited impact. To a large degree this is due to (i) the lack of a flexible and effective signaling infrastructure across the wireless access network, and (ii) the lack of a practical optimization framework that could accommodate signaling delays incurred between different network entities. To address these issues, we propose a five layers backhaul network architecture, named *LayBack*, aimed at extending the notion of software defined networking (SDN) to the dynamic adaptation of wireless resources across different operators and platforms. The *LayBack* (see Sec. II), comprise: the devices layer, the radio node (e.g., eNB, WiFi AP) layer, the gateway layer (e.g., small cell gateways, CRAN), the SDN switching layer, and the SDN backhaul layer (e.g., legacy enhanced packet core (EPC) controlled by SDN applications).

The key idea is extending the benefits of SDN from the realms of the wired backhaul, to the wireless edge, through a unifying SDN orchestrator, providing a flexible and integrated signalling infrastructure that can work across operators and an evolving set of wireless platforms and standards. To fulfill this vision, this paper introduces a formulation of a multi-timescale optimization decomposing the resource allocation into layers so that the orchestrator can centrally control all the resources, while distributing the decision making processes, to quickly and dynamically react to the needs of the network end users. This decomposition also allows the decentralized implementations which is essential to serve 5G dense networks. The proposed framework is inspired by the body of work on Network Utility Maximization (NUM) and the signaling framework of Lagrangian exchanges in standard dual decompositions that we briefly review next.

This work is supported by NSF NeTS: 1716121.

The seminal paper of Kelly et al. in 1997 [8] introduces the concept of NUM to solve the problem of rate allocation in a network with link capacity constraints. This work is followed by extensive amount of efforts which often lie at the intersection between distributed optimization and stochastic network theory; comprehensive surveys can be found in [9]–[11]. A reverse engineering process over former network protocols is also motivated with NUM approach to cast them as optimization problems and gain insights on their efficient performances, or lack thereof. For instance, the work of L. Tassiulas and A. Ephremides [12], [13] on Queue-length Maximum Weight (QMW) scheduling, paved the way for several other researchers who extended the condition under which throughput optimality can be established, or other performance guarantees can be met [14], [15]. In particular, in terms of delay, QMW scheduling is not guaranteed to carry optimal performance [16], leading to the investigation of variations of the algorithm that enhance its delay performances in general multi-hop networks [17] or provide better guarantees [18], [19]. A common feature motivating the decomposition via the NUM formulation is that a centralized optimal scheduler can be impractical, because managing the decoupled networks constraints requires a lot of information. Often, in the decomposition of NUM problems [20], [21] the so-called *time-scale separation assumption* is invoqued, stating that the session interval T_s is much larger than the convergence time T_r of the local resource allocation policy [11]. Under this assumption, in the decomposition one can ignore the convergence of the local control. Several decentralized scheduling algorithms based on queue lengths (e.g. [22]–[24]) rely on this principle.

In this work, we abstract away how the actual physical layer resources (i.e. spectrum and power) granted at the radio node layer (eNB, WiFi AP) provides a dynamic allocation of the rate, and focus on the management of an abstract total rate Z resource at the backhaul, which is indirectly tied to the redistribution of the physical layer channel resources. The SDN operates by keeping the queues logically separated at each eNB, while the shared resource Z trickles down from the orchestrator to the operator, from the operator to the GWs and, finally, from the GWs to the eNBs. In the decomposition of the associated QMW utility over the layers of the

architecture, we consider realistic network latencies, which make the *time-scales separation assumption* unrealistic. The works that remove the *time-scale separation assumption* are divided in two classes: 1) those that use intermediate iterates as decisions and assume continuous underlying flows [25], [26] and 2) those that propose a multi-time scale approach across different layers of the protocol stack [27], [28]. In [25], the authors show that a β-fairness utility function can be maximized, while guaranteeing system stability, under the assumptions that the number of users per class follows a recurrent Markov Chain. We follow a similar rationale as that in [25] for the intermediate decisions. However, since we are not considering a proper utility function, but rather the implicit one that corresponds to the optimal QMW policy, convergence remains an open issue at this point[1]. Like in the second class of works, we consider multi-time scales but those correspond to different layers of the architecture rather than to different allocation problems that take place in different layers of the conventional protocol stack. To illustrate what the Layback architecture could enable, our specific goal in this paper is to focus on the the benefits obtained by sharing the backhaul resources dynamically. In addition to considering different time-scales across the different Layback layers, we also incorporate an economic constraint in the allocation across different operators, which is enforced via the Lyapunov drift plus penalty: a method introduced in [29], [30], and extensively used in recent years for dynamic control. Numerical results suggest that the proposed approach can effectively minimize delays, by enabling a flexible resource redistribution of backhaul resources across different operators.

The rest of the paper is organized as follows. In Section II, we give an overview of the LayBack architecture. In Section III, we present the formulated optimization procedure. In Section IV, we provide numerical examples to illustrate the benefits of the design. The conclusions are in Section V.

II. A brief overview of the LayBack architecture

We briefly review the five layers in the LayBack architecture. The wireless end devices layer encompasses the heterogeneous mobile wireless end devices. The radio access nodes (RAN) layer includes e.g., evolved NodeB (eNB) in LTE or an access point (AP) in Wi-Fi. The gateway (GW) layer encompasses the network entities between the radio node layer and the backhaul (core) entities, e.g., entities of the legacy enhanced packet core. For instance, the GW layer may include the gateways of small cell deployments, or the Base Band Units (BBUs) of a cloud radio access network. The work in [31] introduced the concept of Smart Gateways and discussed the possibility of sharing bandwidth between operators to improve uplink throughput and efficiency. The SDN switching layer consists of SDN switches that flexibly interconnect the RAN layer with the SDN backhaul (core) layer. Radio nodes operating in a non-C-RAN environment (such as macro cell eNBs) process the baseband signals locally and connect directly to the backhaul (core) layer network gateways via the SDN switching layer. The backhaul (core) network layer comprises technology-specific network elements, such as the Evolved Packet Core (EPC) which supports the connectivity of LTE eNBs. The unifying SDN orchestrator in LayBack has three main tasks: 1) it creates a common platform for coordinating among all the wireless service operators and heterogeneous network technologies across its layers; 2) it maintains the current topology information of the entire network and tracks the network capabilities; 3) it enables each of the layers to flexibly reconfigure the network by allocating resources in response to their time-varying needs, while maintaining long term performance requirements that define the service guarantees. This is possible since networks maintained by different operators communicate their requirements and reconfiguration capabilities to the SDN orchestrator. The goal of the remainder of this paper is to showcase how one can embed the NUM decomposition methodology in the SDN centralized management framework.

[1]Even though we have some preliminary results on the convergence properties, these are omitted due to space constraints.

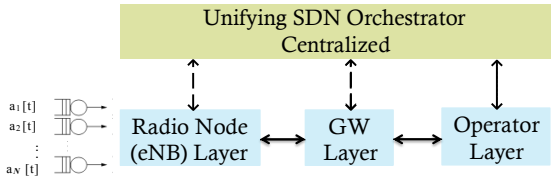


Fig. 1: High level diagram of LayBack architecture. The three illustrated layers are controlled by the SDN orchestrator (Sec. II).

III. Optimization Framework

We consider a network with O distinct operators, indexed by $o = 1, 2, \ldots, O$. Each operator manages a set of Smart-Gateways (GWs) $\mathcal{G}_o$ indexed by $g \in \mathcal{G}_o$. In turn, each GW g manages a set of e-NodeBs (eNBs), indexed by $n \in \mathcal{N}_g$. Let us also define the set $\mathcal{N} \triangleq \bigcup_{o=1}^{O} \bigcup_{g \in \mathcal{G}_o} \mathcal{N}_g$ of all the eNBs and the set $\mathcal{G} \triangleq \bigcup_{o=1}^{O} \mathcal{G}_o$. The queues of each eNB $n \in \mathcal{N}$ are denoted by Q_n and their dynamics are

$$Q_n[t+1] = [Q_n[t] - z_n[t]]^+ + a_n[t+1] \qquad (1)$$

where $a_n[t]$ and $z_n[t]$ represent, respectively, the exogenous packets arrival process and service rate at the backhaul level that is granted to the nth eNB, during the t-th slot. Also, $[\cdot]^+$ denotes projection onto nonnegative orthant ($[\gamma]^+ = \max(\gamma, 0)$). This service rate $z_n[t]$ is a function of the spectrum and power resources allocated for the transmission between t and $t+1$ to the specific eNB as well as its channel state. For now let's assume perfect channel state information is available, an assumption that we will relax later. We define an optimization problem where its input is queue lengths that accumulated in eNBs and output is optimal service rates. Therefore, we do not focus on the physical layer transmission details such as fading, path-loss and noise in the channel between devices layer and eNB layer where optimization is limited with layers shown in Fig. 1.

Before introducing our time scale decomposition, we start from the centralized optimization we wish to emulate, and the

logical steps that decompose the problem in layers via the Lagrange decomposition. If the SDN orchestrator, having full control of the total service rate denoted by Z, could allocate it directly the eNBs, the optimization:

$$\max_{\boldsymbol{z}} \sum_{n\in\mathcal{N}} \mathcal{U}_n(z_n) \text{ s.t. } \sum_{n\in\mathcal{N}} z_n \leq Z,\ 0 \leq z_n \leq Q_n[t]\ \forall n \in \mathcal{N} \tag{2}$$

where we use QMW policy as objective function with $\mathcal{U}_n(z_n) = Q_n[t]z_n$ for the sake of illustrating the decomposition technique. For further implementation, the utility should include channel state w where it should be formulated as $\sum f(Q_n[t], w, z[n])$ when f is a known function of the queue, channel state information and service rate. With the QMW policy, the maximization in (2) leads to the minimization of long term average total queue length, which also results in the minimization of the end-to-end delay in the network (a consequence of Little's theorem [32] for the simplified scenario of continious flows and infinite queue backlogs [33]). There are two issues with solving (2): 1) the idea of having the SDN allocate its network resources at the eNB level does not scale; 2) without any long term constraints, some operators may hoard on backhaul resources. In order to create multiple layers to distribute the decision making processes, we rewrite the maximization in (2) introducing variables that, for the sake of solving (2), are slack variables. As we will see, the additional variables represent actual network decisions in the distributed and time-decomposed implementation of the centralized scheduler. In particular, let us denote by x_o the portion of the wireless service rate Z that is distributed to operator o. Each operator $o = 1, \ldots, O$ redistributes the resources, by giving a portion y_g of x_o to each of its GWs $g \in \mathcal{G}_o$. Similarly, each GW g redistributes the resources, by giving a portion z_n of y_g to each of its eNBs $n \in \mathcal{N}_g$. If all these assignments could happen at the same time-scale indexed by t, distributing the constraints at each layer, the optimization could be solved as follows:

$$\max_{\boldsymbol{x}} \sum_{o=1}^{O} \mathcal{U}_o^\star(x_o; t) \quad \text{s.t.} \sum_{o=1}^{O} x_o \leq Z \tag{3}$$

with $\mathcal{U}_o^\star(x_o; t)$ being the optimal value of the subproblem:

$$\max_{\boldsymbol{y}_o} \sum_{g\in\mathcal{G}_o} \mathcal{U}_g^\star(y_g; t) \quad \text{s.t.} \sum_{g\in\mathcal{G}_o} y_g \leq x_o \tag{4}$$

and $\mathcal{U}_g^\star(y_g; t)$ being the optimal value of

$$\max_{\boldsymbol{z}} \sum_{n\in\mathcal{N}_g} Q_n[t]z_n \text{ s.t. } \sum_{n\in\mathcal{N}_g} z_n \leq y_g,\ 0 \leq z_n \leq Q_n[t]\ \forall n \in \mathcal{N}_g. \tag{5}$$

It is important however to remark that the allocation of $\boldsymbol{x}$ to solve (3) needs to respect an "economic" constraint across the operators, that defines a contractual service obligation and prevents any operator from gaming the system (i.e., consistently acquiring more resources than what it paid for). This constraint on the long-run average of the decisions $\boldsymbol{x}$ is:

$$\limsup_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} x_o[t] \leq Z_o \tag{6}$$

where, for consistency of the problem, it is necessary to have $\sum_{o=1}^{O} Z_o \leq Z$. At the same time, by having an inequality constraint, we are not forced to assign resources to an operator that would be wasted if there is not sufficient uplink demand.

We use the idea of *virtual queues*, following the Lyapunov drift-plus-penalty approach [29] to encode the constraint in (6), and we modify the objective in (3) into:

$$\sum_{o=1}^{O} \mathcal{U}_o^\star(x_o; t) - \frac{1}{V} \sum_{o=1}^{O} \Theta_o[t]x_o. \tag{7}$$

After deciding $\boldsymbol{x}[t]$, the virtual queues Θ_o's are updated as:

$$\Theta_o[t+1] = [\Theta_o[t] + (x_o[t] - Z_o)]^+ \tag{8}$$

where Z_o is the fixed average maximum resource limitation. The parameter V represents the "flexibility" of the constraint in (6), e.g., the higher V the more inclined we are to temporarily violate the constraint. The next subsection serves as a basis to tackle the problem at different timescales, imposed by the network infrastructure, which will be discussed in III-B. It is however easier to derive them in the *ideal* static case first, given that the expressions in the dynamic case will have the same form, albeit having a different meaning.

A. Iterative solution via gradient descent

We will omit the time index t to ease the notation. Dual objective function of the subproblem (3) can be written as

$$\Phi_1\Big(y_g, \lambda_{y_g}; \mathbf{Q}\Big) \triangleq \lambda_{y_g} y_g + \max_{\boldsymbol{z}_g} \sum_{n\in\mathcal{N}_g} (Q_n - \lambda_{y_g}) z_n \tag{9}$$

and we introduce the Lagrangian dual variable λ_Z for the constraint in (3), the Lagrangian dual variables $\{\lambda_{x_o} : o = 1, \ldots, O\}$ for the constraints in (4) and the Lagrangian dual variables $\{\lambda_{y_g} : g \in \mathcal{G}\}$ for the constraints in (5). Then, unfolding all the constraints, we obtain (14) and following a cascade of primal dual decompositions (see [20]), the optimization can be solved via the sequence of projected gradient descent updates:

$$\lambda_Z^{(k+1)} = \left[\lambda_Z^{(k)} - \alpha_1^{(k)}\left(Z - \sum_{o=1}^{O} \operatorname*{argmax}_{x_o} \Phi_4\left(\lambda_z^{(k)}, x_o\right)\right)\right]^+ \tag{10}$$

$$x_o^{(k+1)} = \left[x_o^{(k)} + \alpha_2^{(k)}\left(\operatorname*{argmin}_{\lambda_{x_o}} \Phi_3\left(x_o^{(k)}, \lambda_{x_o}\right) - \lambda_Z - \frac{\Theta_o}{V}\right)\right]^+ \tag{11}$$

$$\lambda_{x_o}^{(k+1)} = \left[\lambda_{x_o}^{(k)} - \alpha_3^{(k)}\left(x_o - \sum_{g\in\mathcal{G}_o} \operatorname*{argmax}_{y_g} \Phi_2\left(\lambda_{x_o}^{(k)}, y_g\right)\right)\right]^+ \tag{12}$$

$$y_g^{(k+1)} = \left[y_g^{(k)} + \alpha_4^{(k)}\left(\operatorname*{argmin}_{\lambda_{y_g}} \Phi_1(y_g^{(k)}, \lambda_{y_g}) - \lambda_{x_o}\right)\right]^+ \tag{13}$$

where α's denote step sizes. The bottom layer optimization in (9) can be solved with the Algorithm 1, while solution for general utility is shown in [34]. We note that to ensure the convergence of the decomposition, the updates in (10)–(13) have to be read as follows: to reach the optimal λ_Z, the SDN orchestrator needs to perform a sufficient number of iterations

$$\overbrace{\min_{\lambda_Z} \lambda_Z Z + \sum_{o=1}^{O} \max_{x_o} \left(-\lambda_Z - \frac{\Theta_o}{V}\right) x_o + \min_{\lambda_{x_o}} \lambda_{x_o} x_o + \sum_{g\in\mathcal{G}_o} \max_{y_g} -\lambda_{x_o} y_g + \min_{\lambda_{y_g}} \Phi_1(y_g, \lambda_{y_g}; \mathbf{Q})}^{\Phi_2(\lambda_{x_o}, y_g; \mathbf{Q})}}^{\Phi_3(x_o, \lambda_{x_o}; \mathbf{Q})}}^{\Phi_4(\lambda_Z, x_o; \mathbf{Q})}}^{\Phi_5(\lambda_Z; \mathbf{Q})} \tag{14}$$

in (10). However, before computing one iteration of (10), the operator layer below should perform a sufficient number of iterations of (11) upon receiving the Lagrangian λ_Z, and so on. Unless a value can be computed in closed form in one shot, each update that includes the solution of an optimization problem (i.e., it has an argmax or argmin term in the update) requires a sufficient number of gradient descent updates at the lower level to approximate the solution of the subproblem. Therefore, the indices k in (10)–(13) are *not* associated with the same time scale. If the computation at each layer and the communication delays among layers were all negligible, we would be in the *time-scale separation* regime. However, this is not possible in a real system, since latencies play an important role and the framework we are about to explain explicitly takes these latencies into consideration. We also note that in this decomposition model, there is no sharing of information among the operators, which makes the model more practical.

Algorithm 1: Solution of (9)

Input : $y_g, \{Q_n : n \in \mathcal{N}_g\}$
Output: $\lambda^\star_{y_g}$
if $\sum_{n\in\mathcal{N}_g} Q_n \geq y_g$ **then**
  Find the permutation $\boldsymbol{\pi} = \{\pi_i : i = 1, \ldots, |\mathcal{N}_g|\}$ to sort the queues $\mathbf{Q}$ such that $i \geq j \Rightarrow Q_{\pi_i} \leq Q_{\pi_j}$;
  Find $i^* = \inf\{i : \sum_{j=1}^{i} Q_{\pi_j} \geq y_g\}$;
  $z_{\pi_j} = Q_{\pi_j}$ for $j < i^*$, $z_{\pi_{i^*}} = y_g - \sum_{j=1}^{i^*-1} Q_{\pi_j}$,
  $z_{\pi_j} = 0$ for $j > i^*$, $\lambda^\star_{y_g} = Q_{\pi_{i^*}}$;
else
  $z_n = Q_n \ \forall n \in \mathcal{N}_g, \lambda^\star_{y_g} = 0$;
end

Let us start by considering the optimization at the bottom layer as the one that operates at the minimum latency, i.e., the time difference between the time indexes t and $t+1$ is the Round Trip Time (RTT) between GW and eNB τ^G_N (considered equal, for simplicity, for all GWs and eNBs), since it is the one closest to devices and to the information regarding traffic. To map all the time instants into integer values of t it is convenient to normalize all times with respect to τ^G_N (i.e., we set $\tau^G_N = 1$). Denoting with $\underline{L}$ and $\underline{P \cdot L}$ the minimum refresh times for the GWs decisions $\boldsymbol{y}$ and for the operators decisions $\boldsymbol{x}$, respectively, time t can be written according to a poly-phase decomposition as

$$t = (mP + p)L + \ell, \ m \in \mathbb{N}, 0 \leq p \leq P-1, 0 \leq \ell \leq L-1$$

where $P \cdot L > \underline{P \cdot L}$ and $L > \underline{L}$ are selected refresh times.

B. Stochastic optimization and temporal decomposition

Since the different layers cannot communicate instantaneously, the parameters of the queues change dynamically underneath. Clearly, the objectives of the optimization have to be defined in such a way that they stay constant while the bottom layer changes stochastically from one state to the other. The proposed framework can be seen as a special case of *stochastic gradient descent* where the network dynamics, via the queues evolution, impose the sequence of training samples' updates. In particular, the SDN orchestrator operates its optimization at every time instant $t = mPL$, performing

$$\begin{aligned} &\max_{\boldsymbol{x}} \sum_{o=1}^{O} -\frac{\Theta_o[m] x_o}{V} + \frac{1}{P} \sum_{p=0}^{P-1} \mathbb{E}\left\{\mathcal{U}^\star_o\left(x_o; (mP+p)\, L\right)\right\} \\ &\text{s.t.} \sum_{o=1}^{O} x_o \leq Z \end{aligned} \tag{15}$$

with $\mathcal{U}^\star_o\left(x_o; (mP+p)\, L\right)$ equal to the optimal value of the problem solved at the operator layer below:

$$\begin{aligned} &\max_{\boldsymbol{y}_o} \sum_{g\in\mathcal{G}_o} \frac{1}{L} \sum_{\ell=0}^{L-1} \mathbb{E}\left\{\mathcal{U}^\star_g\left(y_g; (mP+p)L + \ell\right)\right\} \\ &\text{s.t.} \sum_{g\in\mathcal{G}_o} y_g \leq x_o \end{aligned} \tag{16}$$

and $\mathcal{U}^\star_g\left(y_g; (mP+p)L + \ell\right)$ being the optimal values of the optimization in (5) for $t = (mP+p)L + \ell$. The updates derived in (10)–(13) will then be used to update the decisions $\boldsymbol{x}$ every PL and the decisions $\boldsymbol{y}$ every L, as if convergence to the solution of a static problem has been achieved in the time horizons of length PL and L, respectively. By introducing K_i as the number of iterations of each update in layer $i = 1, \ldots, 4$, respectively starting from the bottom, we can derive the following relations:

$$PL \geq \max\left\{K_4\left(K_3 K_2\left(K_1 + \tau^O_G\right) + \tau^S_O\right), \underline{PL}\right\} \tag{17}$$

$$L \geq \max\left\{K_2\left(K_1 + \tau^O_G\right), \underline{L}\right\}, \tag{18}$$

where τ^S_O and τ^O_G are, respectively, the RTTs between SDN and operators, and between operators and GWs (see also Fig. 2). The inequalities in (17)–(18) indicate that, if we want to act fast, e.g., reduce P and L (possibly to the minimum refresh times) we need to perform fewer iterations. Vice versa, if we want to perform more iterations, we have to be willing to act slower in updating the decisions $\boldsymbol{x}$ and $\boldsymbol{y}$.

In numerical section, we consider a fixed design for P, L, and K_i, $i = 1, \ldots, 4$, and explore the performance. Note that,

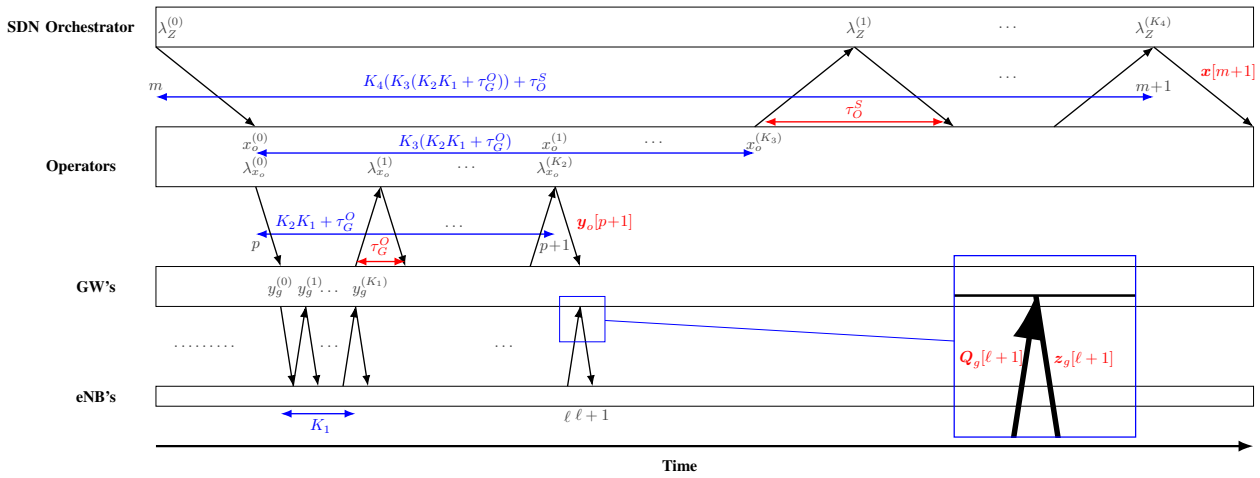
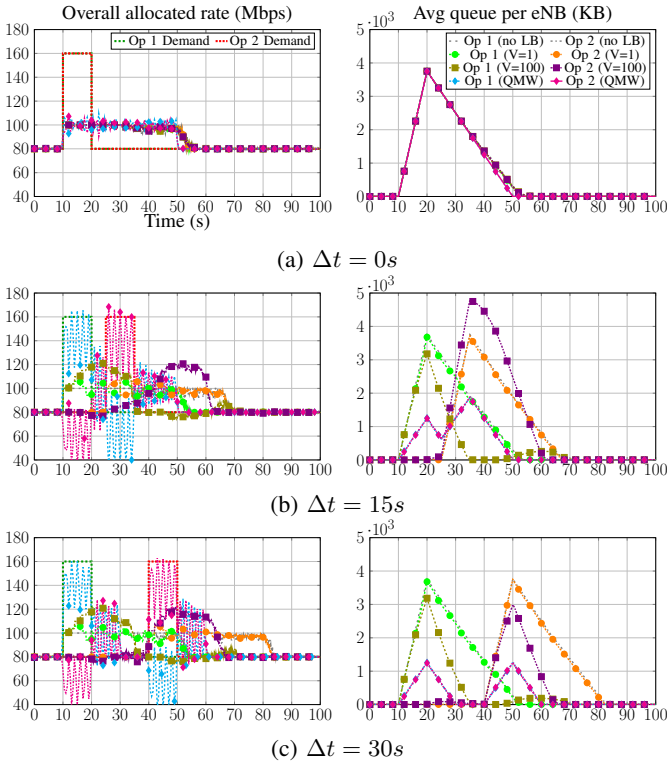


Fig. 2: Illustration of the dynamics of the multi-timescale optimization framework within context of LayBack infrastructure: the optimal policy to minimize end-to-end delay is decoupled into multiple layers of sub-problems, with faster timescale at the lower LayBack layers.

the impact of the choice of the K_i, $i = 1, \ldots, 4$ has not been fully addressed in the literature, where these parameters are implicitly predetermined in the formulations studied. If we look at the static problem, as a "surrogate" for the dynamic problem (up to the next decision), increasing the number of iterations and delaying future decisions can guarantee a better accuracy for a static scenario; however, the ability of the algorithm to incorporate new dynamic information is compromised. That trade-off just described creates another optimization issue which is the subject of our future research and not in the scope of this paper.

IV. Evaluation

In this section, we show the effectiveness of the proposed method in handling demand peaks (i.e., high traffic hours) across different operators by multiplexing resources dynamically. The bottleneck of the proposed approach is that, due to network latencies, high level decisions cannot be instantaneous and if one of the operators experiences a demand peak right after the other, the first of the event creates a response lag in addressing the subsequent events. In our experiments we test different values of the parameter V in (7). Our baselines are: 1) absence of the LayBack orchestrator, e.g. fixed allocation for x_o (labeled "no LB" in the plots) and 2) a centralized optimal scheduler with no latency and no long term constraints limiting operators (labeled "QMW" in the plots). The parameters in Fig. 2 are set to $K_1 = 10, K_2 = 1, K_3 = 5, K_4 = 1, \tau_O^S = 100, \tau_G^O = 10$, which correspond to 1s and 100ms for an RTT between GWs and eNBs of 10ms latency, respectively. L and PL are set to 20 and 200 respectively. For all the updates $\alpha = 0.4$. For numerical stability, the computation of $\lambda_{y_g}^\star$ uses the following queues' normalization $\frac{Q_n}{\sum_{n\in\mathcal{N}_g} Q_n}\frac{|\mathcal{N}_g|}{2}$, which does not alter the solution. The network has the following parameters: $O = 2, |\mathcal{G}_o| = 2\ \forall\ o, |\mathcal{N}_g| = 10\ \forall\ g \in \mathcal{G}, Z_o =$ 100Mbps, $\forall\ o, Z =$ 200Mbps. The aggregate rate demand for each operator is kept constant at 80Mbps, except for a peak of 10s duration of 160Mbps, for each operator. Operator 1 experiences the peak in demand rate at time $t = 10$s, whereas for Operator 2 the peak happens at time $t = (10+\Delta t)$s. At all times, the traffic is homogeneous across the same operator's eNBs. For the selected time parameters and a packet size of 12.5 KBytes, the scenario just described corresponds to a process $a_n[t]$ in (1) as $Pois(0.4)$ in normal conditions and $Pois(0.8)$ when the demand peak occurs. In Fig. 3, we show three different simulations over time for different values of Δt: for $\Delta t = 0$ traffic is perfectly balanced, hence no redistribution across operators is enabled, for $\Delta t = 15$s the aforementioned overshadowing effect can be seen in the delay



(a) $\Delta t = 0s$

(b) $\Delta t = 15s$

(c) $\Delta t = 30s$

Fig. 3: Aggregate rate allocation for the two operators for different values of V and when no sharing across operators is enabled

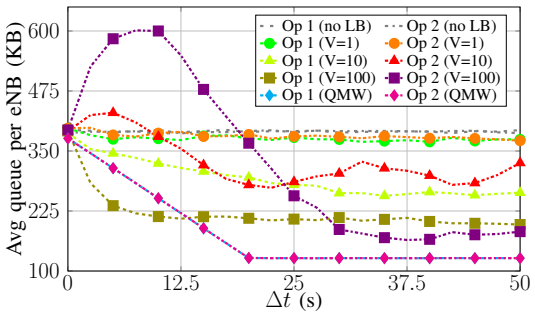


Fig. 4: Average aggregate queue size at each operator for different time distances between demand peaks.

to which the system for $V = 100$ responds to the demand peak for Operator 2. Finally for $\Delta t = 30$s, there is enough time for our decomposition to redistribute the resources and have both operators benefit from sharing. The phenomenon just described is summarized in Fig. 4, where we plotted the time average queue size (over the whole simulation time) vs. the interval Δt that separates the two demand peaks. Notice how for small values of V (e.g $V = 1$) sharing is limited and performances are not significantly different from the absence of SDN orchestration. As V increases, we enable sharing, and when the demand peaks are sufficiently separated, we can guarantee smaller average queues for both operators, closing the gap with the optimal curves for the centralized solution. The shadowing effect described for small Δt is evident since Operator 1 has a smaller average queue size than in the centralized optimization, where it is not prioritized because there is no lag in responding to events occurring later.

V. Conclusions

Leveraging the primal dual decomposition and Lyapunov drift techniques we showed that an SDN centralized management model can be decomposed and become scalable. Numerical experiments validated our approach. This work will be followed by details of the LayBack architecture along with convergence and stability analysis of the algorithm.

References

[1] B. Niu, Y. Zhou, H. Shah-Mansouri, and V. W. S. Wong, "A dynamic resource sharing mechanism for cloud radio access networks," *IEEE T. Wirel. Commun.*, vol. 15, no. 12, pp. 8325–8338, Dec 2016.

[2] T. Biermann, L. Scalia, C. Choi, H. Karl, and W. Kellerer, "CoMP clustering and backhaul limitations in cooperative cellular mobile access networks," *Pervasive and Mobile Comp.*, vol. 8, pp. 662–681, 2012.

[3] T. Liu, K. Wang, C. Ku, and Y. Hsu, "QoS-aware resource management for multimedia traffic report systems over LTE-A," *Computer Networks*, vol. 94, pp. 375–389, Jan. 2016.

[4] T. Taleb, Y. Hadjadj-Aoul, and K. Samdanis, "Efficient solutions for enhancing data traffic management in 3GPP networks," *IEEE Systems Journal*, vol. 9, no. 2, pp. 519–528, 2015.

[5] J. Gutiérrez and et al., "5G-XHaul: a converged optical and wireless solution for 5G transport networks," *Trans. ETT*, vol. 27, no. 9, pp. 1187–1195, 2016.

[6] H. Liu, H. Zhang, J. Cheng, and V. C. Leung, "Energy efficient power allocation and backhaul design in heterogeneous small cell networks," in *Proc. IEEE ICC*, 2016, pp. 1–5.

[7] A. De Domenico, V. Savin, and D. Ktenas, "A backhaul-aware cell selection algorithm for heterogeneous cellular networks," in *IEEE Personal Indoor and Mobile Radio Commun.*, 2013, pp. 1688–1693.

[8] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[9] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Area. Comm.*, vol. 24, no. 8, pp. 1452–1463, 2006.

[10] M. Chiang, S. H. Low, R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proc. IEEE*, vol. 95, pp. 255–312, 2007.

[11] M. Chiang, "Stochastic network utility maximization," *Eur. T. on Telecommun.*, vol. 22, pp. 1–22, 2008.

[12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE T. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1948, 1992.

[13] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE T. Inform. Theory*, vol. 39, no. 2, pp. 466–478, 1993.

[14] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," *IEEE T. Wirel. Commun.*, vol. 7, no. 7, 2008.

[15] B. Ji, C. Joo, and N. Shroff, "Throughput-optimal scheduling in multihop wireless networks without per-flow information," *IEEE/ACM T. Netw.*, vol. 21, no. 2, pp. 634–647, 2013.

[16] Y. Cui and E. M. Yeh, "Delay optimal control and its connection to the dynamic backpressure algorithm," in *IEEE Int. Symp. Info.*, 2014, pp. 451–455.

[17] Y. Cui, E. M. Yeh, and R. Liu, "Enhancing the delay performance of dynamic backpressure algorithms," *IEEE/ACM T. Netw.*, vol. 24, no. 2, pp. 954–967, 2016.

[18] K. Kar, S. Sarkar, A. Ghavami, and X. Luo, "Delay guarantees for throughput-optimal wireless link scheduling," *IEEE T. Automat. Contr.*, vol. 57, no. 11, pp. 2906–2911, Nov 2012.

[19] M. J. Neely, "Delay-based network utility maximization," *IEEE/ACM T. Netw.*, vol. 21, no. 1, pp. 41–54, 2013.

[20] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Area. Comm.*, vol. 24, no. 8, pp. 1439–1451, Aug 2006.

[21] B. Johansson, P. Soldati, and M. Johansson, "Mathematical decomposition techniques for distributed cross-layer optimization of data networks," *IEEE J. Sel. Area. Comm.*, vol. 24, pp. 1535–1547, 2006.

[22] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *IEEE/ACM T. Netw.*, vol. 17, no. 6, pp. 1846–1859, 2009.

[23] L. X. Bui, S. Sanghavi, and R. Srikant, "Distributed link scheduling with constant overhead," *IEEE TON*, vol. 17, no. 5, pp. 1467–1480, 2009.

[24] Y. Teng and M. Song, "Cross-layer optimization and protocol analysis for cognitive ad hoc communications," *IEEE Access*, 2017.

[25] X. Lin, N. B. Shroff, and R. Srikant, "On the connection-level stability of congestion-controlled communication networks," *IEEE T. Inform. Theory*, vol. 54, no. 5, pp. 2317–2338, 2008.

[26] R. Srikant, "On the positive recurrence of a markov chain describing file arrivals and departures in a congestion-controlled network," in *IEEE Conf. Comput.*, 2004.

[27] E. Altman, K. Avrachenkov, and S. Ramanath, "Multiscale fairness and its application to resource allocation in wireless networks," *Comput. Commun.*, vol. 35, no. 7, pp. 820–828, 2012.

[28] Q.-V. Pham, H.-L. To, and W.-J. Hwang, "A multi-timescale cross-layer approach for wireless ad hoc networks," *Comput. Netw.*, vol. 91, pp. 471–482, 2015.

[29] 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, Apr. 2006.

[30] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE T. Inform. Theory*, vol. 52, pp. 2915–2934, Jul. 2006.

[31] A. Thyagaturu, Y. Dashti, and M. Reisslein, "SDN based smart gateways (Sm-GWs) for multi-operator small cell network management," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 740–753, 2016.

[32] A. O. Allen, "Statistics and queuing theory with computer science applications, vol. 2," 1990.

[33] R. Banirazi, E. Jonckheere, and B. Krishnamachari, "Heat diffusion algorithm for resource allocation and routing in multihop wireless networks," in *IEEE Glob. Commun. Conf.*, 2012, pp. 5693–5698.

[34] S. H. Low and D. E. Lapsley, "Optimization flow control. i. basic algorithm and convergence," *IEEE TON*, vol. 7, pp. 861–874, 1999.