Dynamic Network Reconfiguration in Wireless DenseNets with the CROWD SDN Architecture

Sébastien Auroux and Martin Dräxler
University of Paderborn
Paderborn, Germany
E-mail: sebastien.auroux@upb.de
martin.draexler@upb.de

Arianna Morelli
INTECS
Pisa, Italy
E-mail: arianna.morelli@intecs.it

Vincenzo Mancuso
IMDEA Networks Institute
Madrid, Spain
E-mail: vincenzo.mancuso@imdea.org

***Abstract*—Traffic in wireless access networks has been growing substantially in the recent years, both in terms of total volume and data rate required by individual users. The commonly agreed solution to overcome the current limitations of wireless access networks is to deploy very dense and heterogeneous wireless networks, the so-called *DenseNets*. Simply scaling existing networks by orders of magnitude, as required to fulfill traffic forecasts, would bring along several problems because of the limited backhaul capacity, the increased energy consumption, and the explosion of signalling. Hence, the FP7 project CROWD proposes a novel architecture for DenseNets as a solution to tame the density of wireless networks. Within this architecture, *flexible flow processing-aware controller placement* supported by *dynamic backhaul reconfiguration* is the crucial component to both (i) maximize the amount of users that can be served simultaneously under high load, and (ii) minimize energy consumption and reduce costs for service providers. We present both approaches in detail and outline how they are integrated into the overall CROWD architecture.**

I. Introduction

Wireless traffic demands have been growing exponentially in recent years and this trend is expected to continue. It is commonly agreed that this growing demand can only be satisfied by increasing the density of points of access and combining different wireless technologies. While mobile network operators already push for denser networks and heterogeneous deployments [1], the efficiency of the current technology needs to be improved as well, to prevent unsustainable costs for energy consumption and hardware extensions.

In previous work [2] we identified Software-Defined Networking (SDN) as a promising choice to face this task. The SDN approach separates the network data plane from the control plane and limits the data plane or the switches to be responsible only for traffic forwarding. Instead, the network behavior is controlled from a single, high-level control application, the *controller*. This permits a network operator to run more complicated control programs for the data paths via a well-defined control-plane interface. Also, the data plane and the control plane are able to evolve and to be developed independently. In particular, the software control of the network can evolve independently of the hardware.

The CROWD project[1] proposes a novel wireless network architecture, extending the concept of SDN, with advanced control applications that perform management operations, such as traffic load balancing or dynamic allocation of network resources. Like in legacy SDN approaches, CROWD makes control explicit; however, CROWD *goes beyond SDN by enabling control applications to process data flows*. The controllers running such applications need to operate on different scopes and time scales: short time scale (frame transmission duration), medium time scale (flow lifetime), and long time scale (power cycling of network elements). Thus a deliberate placement and flexible reassignment of the controllers and an adequate assignment of the backhaul resources are the crucial factors to enable these applications and to avoid, at any scope and time scale, to incur in bottlenecks within the network. In our previous work [2]–[4] we have introduced the overall architecture and SDN-oriented concepts of CROWD. In this paper we specially focus on the novel *flexible flow processing-aware controller placement* and *dynamic backhaul reconfiguration* components and on *their combination*, that is crucial for the performance of the overall CROWD architecture. In addition, we provide examples of key control applications that benefit from the unique features offered by our dynamic mechanisms in the framework of the CROWD architecture.

The rest of the paper is structured as follows. Section II discusses related work. Section III introduces the CROWD controller framework, with special focus on flexible controller placement and dynamic backhaul reconfiguration. Section IV gives an overview of the framework's implementation, and Section V summarizes and concludes the paper.

II. Related Work

The controller placement for SDNs is already a widespread topic in current research [5]–[7]. The authors of [5] try to maximize the expected percentage of valid control paths with their controller placement model, achieving very decent results with heuristic algorithms on real-world ISP topologies. References [6], [7] take an additional step by taking into account controller reassignment. Bari *et al.* [6] present a placement model with periodic controller reassignment, while the authors of [7] propose an distributed controller architecture adjusting the number of used controllers based on the workload of the individual controllers.

None of the aforementioned work considers the stringent data rate and processing capacity requirements demanded by sophisticated control applications in future wireless dense networks. Moreover, controller reassignment should be initialized

[1] http://www.ict-crowd.eu

dynamically in reaction to the current network state, to continuously guarantee high efficiency of the control applications in the network. Our flexible flow processing-aware controller placement model takes all this into account in addition to constraints imposed by conventional SDN control aspects that have already been studied by other authors to a certain extent.

The effects of a static, limited backhaul network on wireless cooperation techniques have been investigated in [8], [9], as well as the gain from using a dynamic, reconfigurable backhaul network has been investigated in [10]. However, these approaches do not apply the concept of dynamic backhaul network reconfiguration on a whole controller framework for wireless coordination, and cannot be used in large and very dense networks experiencing highly variable offered load, like the networks studied in CROWD.

III. CROWD Controller Framework

In CROWD, we consider very dense and heterogeneous wireless networks, the so-called *DenseNets*. We argue that having one static, centralised controller is not an adequate solution taking into account the density of the network and the expected computational overhead. Controllers have to be able to manage a broad range of dynamics happening at very different time scales, due to the heterogeneity of DenseNets. The envisioned control applications range from very fast, short time scale functions, executed close to the involved network equipment, to long-term functions involving larger parts of the backhaul network. Moreover, controllers need to aggregate network information as an input for control applications, causing signalling overhead. Therefore, CROWD proposes a dynamic, two-tier SDN controller hierarchy with two types of controllers:

- the *CROWD Local Controller (CLC)* makes fast, short time scale decisions on a limited but fine-grained scope;
- the *CROWD Regional Controller (CRC)* makes slower, medium-long time scale decisions with a broader but coarse-grained scope.

CLC/CRC functionalities are special applications, which can be installed on network equipment with hardware fulfilling the necessary requirements, i.e., sufficient processing capacity and memory, on demand. This two-tier approach allows to better aggregate network information when details are not required on a broader scope and reducing signaling overhead.

The CROWD architecture is structured into two logical tiers called *district* and *region*, being controlled by a CLC and a CRC respectively. An overview of the entire CROWD Controller Framework is illustrated in Figure 1. A *district* consists of base stations, i.e., LTE eNBs or WiFi APs, as well as interconnecting backhaul links that are assumed, without loss of generality, to be reconfigurable via some protocol, e.g., OpenFlow [11]. Operation within a district is optimized by applications connected to the CLC via a set of APIs, called North-Bound interface in SDN terminology. Any application can connect to one or more APIs, depending on its optimization goals and requirements.

The North-Bound interface also connects a CLC to its higher-level controller, the CRC, which operates inside a *region*. The region is defined as a logical area including several

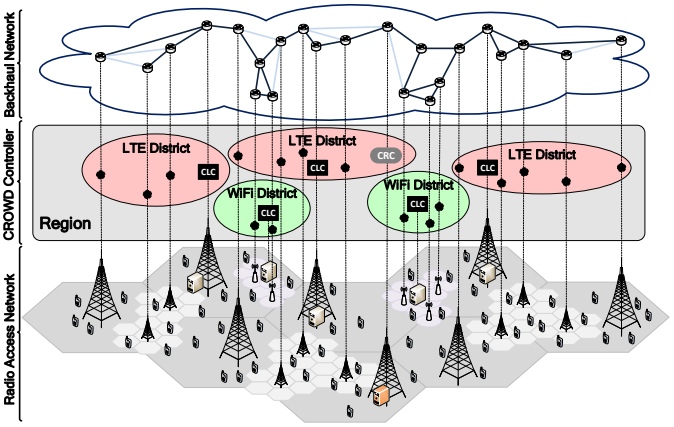


Figure 1. CROWD Controller Framework.

districts in which applications are executed for longer time scale optimizations. Regional optimization is necessary to compensate for sub-optimal choices which may be taken at district level because of the myopic view of the local controllers. For example, from the point of view of an application running with the goal of minimizing energy consumption in a district, an optimal choice could be forcing all user terminals to associate to access points outside the district and switching off all the network elements. Such a drastic decision would be obviously sub-optimal from a broader network viewpoint, thus any regional application aiming to minimize energy consumption would certainly override it.

To allow an efficient operation of this controller framework, the controllers cannot be placed statically but should be placed, instantiated, and reassigned dynamically. Also the backhaul network has to be reconfigured dynamically to accommodate all demands from changing controller locations. In Sections III-A and III-B we outline the *compound* formal requirements for controller placement and backhaul reconfiguration. Both mechanisms are dynamically and jointly performed whenever the network state changes. We further elaborate on other potential CROWD control applications in Section III-C.

A. Flexible Flow Processing-aware Controller Placement

The control applications in CROWD include optimization strategies and mechanisms that require fast and reliable communication between different network elements. Depending on the specific application, the communication has to follow stringent constraints in terms of required data rate and latency for the network connection and in terms of processing capacity for the designated controllers. In our model, we generalize these applications by abstracting them as data flows coming from one or more nodes of the network. These flows require certain data rates and latencies to a CLC to be processed and coordinated, taking into account a given processing capacity needed. This abstract generalization incorporates a large variety of network mechanisms, such as Coordinated Multi-Point transmission (CoMP) being coordinated at a CLC or even a Virtual Network function deployed at a CLC. Hence, the CROWD controller placement turns into a non-trivial, NP-hard problem [12].

We consider the cellular network represented as as graph $G = (V, E)$ with nodes V, e.g. LTE eNBs, WiFi APs and switches, and edges E, representing the backhaul links of

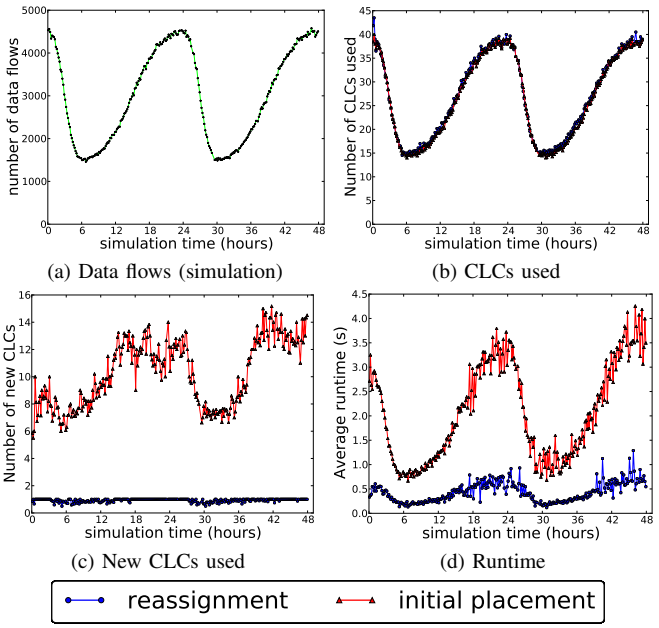
the network. Every link $(u, v) \in E$ is unidirectional with latency $l_{\text{cap}}(u, v)$ seconds and maximum data rate $b_{\text{cap}}(u, v)$ bit/s. The potential controllers are denoted by $C \subset V$ with a processing power of $p_{\text{node}}(c)$ FLOPS. Moreover, we consider a set F of flows passing through one or more nodes from V and which require processing at a CLC. The connection matrix $W \in \{0, 1\}^{|F|\times|C|}$ indicates which node a given flow is passing through. Each flow $x \in F$ requires a data rate of $b_{\text{flow}}(x)$ bit/s from the routing path from its nodes and the assigned CLC. Furthermore, a maximum round-trip latency of $l_{\text{flow}}(x)$ seconds is required, taking into account a necessary processing capacity of $p_{\text{flow}}(x)$ FLOPS.

Each node $v \in V$ is required to be controlled by at least one CLC and each CLC is required to be assigned to exactly one CRC for coordination purposes. Being the CLC of a node or the CRC of a CLC requires a processing capacity of p_{CLC} or p_{CRC} respectively. Furthermore, the connection between a node and its CLC or a CLC and its CRC needs to meet a latency of l_{CLC} seconds or l_{CRC} seconds respectively, as well as a data rate of b_{CLC} bit/s or b_{CRC} bit/s. Our model calculates the corresponding routing paths to ensure that these constraints are fulfilled. We use binary variables $f_{c,d,u,v}$ to express that $(u, v) \in E$ is used on the routing path from node $d \in V$ to CLC $c \in C$. Similarly, the binary variables $g_{c,d,u,v}$ are used to describe the routing paths between a CLC and its CRC.

Apart from implementing the control hierarchy as illustrated in Figure 1, our main objective is to *satisfy* each data flow $x \in F$ by one CLC $c \in C$, i.e., c controls all nodes v with $W_{x,v} = 1$ and all routing paths from the nodes v with $W_{x,v} = 1$ to c provide at least the additional data rate $b_{\text{flow}}(x)$ and cope with the requested latency $l_{\text{flow}}(x)$, paying regard to the required processing capacity $p_{\text{flow}}(x)$ for processing x at c.

In [12], [13] we have presented optimization models and a heuristic to solve the flexible flow processing-aware controller placement problem. But since, our heuristic has been massively improved and is now also able to flexibly modify the current controller placement in reaction to the current network load. Although, it does not consider backhaul reconfiguration issues (see Section III-B).

Figure 2 illustrates new evaluation results from a dynamic 48 hour simulation performed on a network with 100 nodes and mesh backhaul topology with data flows whose parameters require operation on different time scales (for more details on the topology and parameters used, please refer to [13]). In this simulation, we have compared our flexible controller reassignment with an *initial* controller placement, i.e., a placement performed from scratch each time the set of CLCs had to be modified. For all graphs, we have taken the averages for 10 minute intervals for a better overview. During the whole simulation, only one CRC has been used and all data flows were satisfied at any time. Figure 2a depicts the amount of data flows in our simulation. As can be seen in Figure 2b, both the flexible reassignment and the initial controller placement adapt very well to the amount of data flows in the network, while the flexible reassignment only uses negligibly more CLCs on average than the initial controller placement. Figure 2c shows the amount of newly added CLCs compared to the previous controller placement. As can be seen, the flexible reassignment builds up very well on the previous placement, while the initial controller placement determines many new CLCs without



(a) Data flows (simulation) (b) CLCs used

(c) New CLCs used (d) Runtime

Figure 2. Flexible flow processing aware controller placement evaluation

knowing about the previous placement. As a result, the flexible reassignment saves a lot of reconfiguration overhead compared to a network reconfiguration achieved by simply calculating a new controller placement from scratch. Last but not least, Figure 2d illustrates that the flexible reassigment reacts significantly faster to network load changes than the initial controller placement, especially in high load situations. In crowded scenarios, we see a small runtime as a crucial advantage.

B. Dynamic Backhaul Reconfiguration

With the above, we have introduced capacity and latency constraints for the data exchange between network elements (LTE eNBs or WiFi APs) and controllers over the backhaul network. However, controller placement does not guarantee optimal network performance if the network cannot be reconfigured. Therefore, we look at performance constraints in more detail while considering a dynamic, reconfigurable backhaul network.

To implement a reconfigurable backhaul network, the transport technology of the backhaul network needs to be able to change the characteristics (e.g., the capacity) of network links. This is possible in wavelength division multiplexing passive optical networks (WDM-PONs). WDM-PON [14] is an optical fibre technology that multiplexes optical carrier signals on an optical fibre using different wavelengths or colours of laser light. This allows great flexibility in deploying and operating backhaul networks: wavelengths can be added, dropped or manipulated in network nodes, allowing to build dynamic topologies in the backhaul network. Due to this ability to dynamically reconfigure the backhaul network, by enabling and disabling wavelengths on links, we consider WDM-PON as the backhaul technology for this paper. Of course our approach also applies to any other backhaul technology that provides a similar level of flexibility.

With WDM-PON, instead of assuming a static maximum data rate per link $b_{cap}(u, v)$, we consider that each link has a set of wavelengths K with a maximum data rate $b_{cap}(u, v, k)$ for each wavelength $k \in K$. This data rate is of course only

available if the wavelength k is currently assigned to the link (u, v), which we denote as $a_{u,v,k} \in \{0, 1\}$. Furthermore, each routing path has to be assigned to one wavelength per edge that it traverses. We denote this assignment by adding another index to the flow variables introduced in Section III-A, i.e., $f_{c,d,u,v,k} \in \{0, 1\}$ for the routing paths from nodes to CLCs. With these variables we can define several constraints, depending on the technological implementation of the backhaul network. For WDM-PON networks these constraints are:

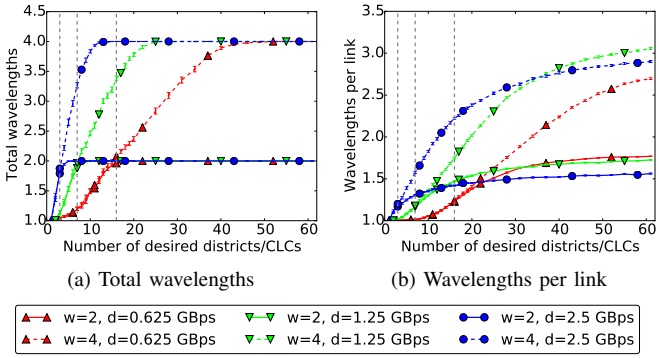
- If the optical fibres are used bidirectionally, each wavelength can only be assigned in one direction: $a_{u,v,k} + a_{v,u,k} \leq 1$
- If no wavelength conversion should be applied to a wavelength k at node v on incoming edge (u, v), wavelength k may only be assigned to exactly one outgoing edge (v, w), every CLC routing path assigned with $f_{c,d,u,v,k}$ has to be assigned with $f_{c,d,v,w,k}$ and no other routing path with $f_{c',d',u',v,k}$ can be assigned.
- Similarly, a full optical wavelength conversion can be defined. If wavelength k at node v on incoming edge (u, v) should be converted to wavelength j on outgoing edge (v, w), every CLC routing path assigned with $f_{c,d,u,v,k}$ has to be assigned with $f_{c,d,v,w,j}$.

If the dynamic assignment of CLCs and CRCs changes, also the data flows and data rate demand change. In order to react to these changes and ensure feasibility of controller assignment, the wavelength and flow assignment have to be dynamically adapted within the backhaul network. In our previous work [15] we have presented both a complete version of this optimization problem and a heuristic algorithm to implement *dynamic radio and backhaul configuration*. We have shown how we can both increase the feasibility of enhanced wireless mechanisms and reduce required backhaul resources, although in there we did not consider the controller placement issue described above.

Figure 3 summarizes our evaluation from previous work [15] with a scenario of 36 base stations and a mesh topology for the backhaul (for more details on the evaluation scenario, please refer to [15]): We compare the cases of having either 2 or 4 wavelengths available per link to assign in the backhaul network (parameter k) and different capacity demands per district (parameter d). Looking at the average number of assigned wavelengths in the backhaul in total (Figure 3a) and per link (Figure 3b) we can see that up to the turning points (dashed vertical lines), the *dynamic radio and backhaul configuration* mechanism assigns the same number of wavelengths for both 2 and 4 available wavelengths, confirming that the mechanism only assigns additional wavelengths if necessary. Also, after the turning points the number of assigned wavelengths increases gradually and does not escalate immediately. We can thus conclude that the mechanism dynamically assigns wavelengths precisely according to the current demand.

C. Other CROWD Control Applications

To connect the above mentioned control applications by means of a control architecture is the scope of CROWD. However, it is definitely not trivial to connect these aspects between them and with other applications that have to serve at different time scales and scopes. To exemplify the complexity of such integration, here we provide three additional representative



(a) Total wavelengths (b) Wavelengths per link

Figure 3. Dynamic radio and backhaul configuration evaluation

examples of the other potential control applications for different time scales that can be implemented in the CROWD architecture and are currently under investigation in the project. We will comment on implementation techniques in Section IV.

ABFS control: This short time scale application represents a case of MAC enhancement in which ABSF patterns within an LTE district are controlled to dynamically reduce the downlink and uplink interference [16]. ABSF adaptation occurs based on the channel quality reports sent by the UEs (downlink) and on the physical measurements taken by the eNBs (uplink), as well as on the current traffic load and QoS constraints, if any. Data is collected via the X2 interface [16] and is sent to the CLC. In turn, using it's APIs, the CLC filters per-user data and provides aggregated per-eNB statistics to the control application. The control application periodically computes the appropriate ABSF patterns, one for each eNB, and sends them back to the CLC, who dispatches the patterns to eNBs.

User association control. This medium time scale application involves both access and backhaul, since UEs are driven in the process of selecting an eNB in a district. The controller collects network utilization statistics and channel qualities using X2, as to maximize either capacity or energy efficiency under backhaul network constraints. Energy efficiency is achieved by aggregating UEs into as few eNBs as possible (and also possibly selecting those eNBs which are connected to the same backhaul network elements) so that unused devices can be eventually switched off. The algorithm running in the control application instructs the CLC with the optimal UE-eNB association map, and eNB-initiated handovers are enforced if needed.

Power cycling control. This longer time scale application focuses on resource-on-demand and energy efficiency in the network infrastructure. To achieve these goals, CLC and CRC take power cycling patterns of base stations, access points, and backhaul network elements to minimize energy consumption, based on the estimated utilisation and traffic load. The CLCs are responsible for the power cycling of access elements while CRCs are responsible for the reconfiguration of backhaul resources in accordance with the power cycling enforced by the CLCs. In this case, either X2 [16] or CAPWAP [17] is used to switch on or off wireless modules of eNBs or WiFi access points, respectively. In addition, CRCs use OpenFlow [11] to reconfigure the backhaul, whereas CLCs and CRCs use their APIs to coordinate their efforts.

IV. Implementation of CROWD Controllers

In the previous section, we have introduced the concepts of flexible flow processing-aware controller placement and dynamic backhaul reconfiguration. In general, with the CROWD architecture, we aim to provide the tools to control the operation of the radio access network and the reconfiguration of backhaul and access resources by means of CRCs and CLCs. To achieve this goal, in addition to controllers, we have defined the protocols to be used in the *North-Bound* and *South-Bound* interfaces of the controllers.

Control Protocols. The basic protocols required to connect controllers and network access elements (South-Bound) are extended versions of *X2* [16] for LTE and *CAPWAP* [17] for WiFi. These protocols are technology-specific and include flexible capabilities. These two features make it possible to extend the set of messages and commands that can be managed. As concerns backhaul elements, the controllers use *OpenFlow* [11] to reconfigure links and routing (South-Bound). Control applications communicate with controllers on the North-Bound interface by using a protocol based on the *REST* paradigm, which is technology-agnostic and can be used to control both LTE and WiFi network components.

Open Source Controllers. CROWD controllers are implemented as an extension of the OpenDaylight open source Java-based platform [18]. OpenDaylight itself only provides controller functionality for OpenFlow. For integrating it in a heterogeneous dense wireless network we extend the framework by new South-Bound interfaces for LTE and WiFi.

Interfaces. Furthermore, North-Bound interfaces, implemented as REST-APIs, provide control over the network elements. Existing REST-APIs are already available, but we also extend these interfaces to specific functionalities of the CLC and CRC. Some of the existing OpenDaylight components are used, i.e., the topology manager, the switch manager (providing the management of the network elements) and the Service Abstraction Layer (SAL) that is the key element to be able to translate information coming from the South-Bound interfaces into parameters which have to be analysed, filtered and sent to the North-Bound interface.

V. Conclusion and Outlook

We have described the framework for flexible flow processing-aware controller placement and dynamic backhaul network reconfiguration in CROWD. Our approach endorses and enhances the principles of SDN and uses the dynamic reconfigurability of modern network elements. Our approach is capable of efficiently operating the control applications for future wireless DenseNets in presence of heterogeneous control applications with different scopes and time scale requirements.

Our current research has already generated fast heuristic algorithms with close to optimal results for both flow processing-aware controller placement and dynamic backhaul network reconfiguration, although the complexity of integrating such components with other control applications in DenseNets had not been so far considered before this paper. Future work will lead to the design of a prototype implementation of a control integrated framework for CLCs and CRCs.

Acknowledgment

The research leading to these results has received funding from the EU Seventh Framework Programme (FP7/2007-2013) under grant agreement 318115 (Connectivity management for eneRgy Optimised Wireless Dense networks – CROWD).

References

[1] P. Bhat, S. Nagata, L. Campoy, I. Berberana, T. Derham, G. Liu, X. Shen, P. Zong, and J. Yang, "LTE-Advanced: An operator perspective," *IEEE Communications Magazine*, pp. 104–114, 2012.

[2] H. Ali-Ahmad, C. Cicconetti, A. de la Oliva, M. Dräxler, R. Gupta, V. Mancuso, L. Roullet, and V. Sciancalepore, "Crowd: An sdn approach for densenets," in *2nd European Workshop on Software Defined Networks (EWSDN)*, 2013.

[3] C. Cicconetti, A. Morelli, M. Dräxler, H. Karl, V. Mancuso, V. Sciancalepore, R. Gupta, A. de la Oliva, I. Snchez, P. Serrano, and L. Roullet, "The playground of wireless dense networks of the future," in *Future Network and MobileSummit (FuNeMS)*, 2013.

[4] H. Ali-Ahmad, C. Cicconetti, A. de la Oliva, V. Mancuso, M. R. Sama, P. Seite, and S. Shanmugalingam, "An sdn-based network architecture for extremelydense wireless networks," in *IEEE Software Defined Networks for Future Networks and Services (IEEE SDN4FNS)*, 2013.

[5] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2013, pp. 672–675.

[6] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic controller provisioning in software defined networks," in *9th IEEE/ACM/IFIP International Conference on Network and Service Management (CNSM)*, 2013.

[7] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed sdn controller," in *2nd ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013.

[8] T. Biermann, L. Scalia, C. Choi, H. Karl, and W. Kellerer, "Backhaul Network Pre-Clustering in Cooperative Cellular Mobile Access Networks," in *IEEE World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2011.

[9] O. Simeone, O. Somekh, H. V. Poor, and S. Shamai, "Downlink multicell processing with limited-backhaul capacity," *EURASIP Journal on Advances in Signal Processing*, 2009.

[10] M. Dräxler, T. Biermann, and H. Karl, "Improving cooperative transmission feasibility by network reconfiguration in limited backhaul networks," *International Journal of Wireless Information Networks*, vol. 20, no. 3, pp. 183–194, 2013.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[12] S. Auroux and H. Karl, "Flow processing-aware Controller Placement in Wireless DenseNets," in *25th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2014.

[13] ——, "Efficient flow processing-aware controller placement in future wireless networks," in *IEEE Wireless Communications and Networking-Conference (WCNC)*, 2015.

[14] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective*, ser. Morgan Kaufmann. Elsevier/Morgan Kaufmann, 2010.

[15] M. Dräxler and H. Karl, "Feasibility of base station coordination and dynamic backhaul network configuration in backhaul networks with limited capacity," in *European Wireless 2014 (EW2014)*, 2014.

[16] Third Generation Partnership Project (3GPP), "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 application protocol (X2AP), 3rd ed." 3GPP, technical Specification Group Radio Access Network.

[17] P. Calhoun, M. Montemurro, D. Stanley *et al.*, "Control and provisioning of wireless access points (CAPWAP) protocol binding for IEEE 802.11," *IETF RFC5416*, 2009.

[18] "Opendaylight: http://www.opendaylight.org/."