Software-Defined Networking Enabled Capacity Sharing in User Centric Networks

Bruno Astuto Nunes, Mateus Augusto Silva Santos, Bruno Trevizan de Oliveira, Cintia Borges Margi, Katia Obraczka, Thierry Turletti

Software-Defined Networking Enabled Capacity Sharing in User Centric Networks

Bruno A. A. Nunes, Mateus A. S. Santos, Bruno T. de Oliveira, Cintia B. Margi, Katia Obraczka, and Thierry Turletti

***Abstract*—In this paper, we discuss User Centric Networks (UCNs) as a way of, if not completely solving, considerably mitigating the problem of sharing limited network capacity and resources efficiently and fairly. UCNs are self-organizing networks where the end-user plays an active role in delivering networking functions such as providing Internet access to other users. We propose to leverage the recently proposed Software Defined Networking (SDN) paradigm to enable cooperation between wireless nodes and provide capacity sharing services in UCNs. Our SDN-based approach allows to extend coverage of existing network infrastructure (such as WiFi or 3GPP) to other end-users or ad hoc networks that would otherwise not be able to have access to network connectivity and services. Moreover, the proposed SDN-based architecture also takes into account current network load and conditions, and quality-of-service (QoS) requirements. Another important feature of our framework is that security is an integral part of the architecture and protocols. We discuss the requirements for enabling capacity sharing services in the context of UCNs (e.g., resource discovery, node admission control, cooperation incentives, QoS, security, etc) and how SDN can aid in enabling such services. The paper also describes the proposed SDN-enabled capacity sharing framework for UCNs.**

***Index Terms*—Software-Defined Networking, programmable networks, user centric networks, capacity sharing, user as a provider, load balancing, fault tolerance, node admission control.**

I. Introduction

THE 1990's futuristic vision of "ubiquitous computing" and "anywhere, anytime connectivity" is now, only 20 years later, a reality, enabled mostly by widespread access to both portable computing devices as well as wireless communication infrastructure. Over the past few years, anywhere, anytime connectivity has resulted in exponential increase in mobile traffic, which is expected to outgrow the capabilities of current 4G and LTE infrastructure in the near future.

One possible solution to this problem would be, of course, to provision and upgrade the network infrastructure, for example, by deploying a higher number of more capable access points and base stations (e.g., conventional macro base stations or pico/femto-cells). However, "throwing bandwidth at the problem", i.e., augmenting network infrastructure at the same rate as traffic demand increases comes with considerably high-, and most of the time, prohibitive costs.

Consequently, a major challenge facing future networks is to provide ubiquitous connectivity in a scalable and resource-efficient fashion. This problem has been referred to as "network capacity sharing" [1] and has drawn considerable attention from industry and academia. "*User-Centric Networks*", or UCNs, have emerged as a way of, if not completely solving, considerably mitigating the problem of sharing limited network capacity and resources efficiently and fairly. UCNs are self-organized networks where the end-user plays an active role in networking functions such as providing Internet access to other users. As such, in UCNs, end users can act as "micro network operators" sharing their subscribed Internet access with other users often based on some incentive mechanism. Besides extending the coverage of the Internet's backbone infrastructure at marginal cost mitigating the capacity sharing dilemma, UCNs also improve communication services, fault tolerance and detection, as well as load balancing. On the other hand, UCNs raise a wealth of interesting challenges of their own ranging from providing adequate security and trust management, incentivizing users to act as micro network operators, understanding and harnessing user mobility, and coping with intermittent connectivity, to name a few.

In this article, we explore Software-Defined Networking (SDN) as a promising approach to address some of the challenges raised by UCNs, in particular providing efficient network capacity sharing services. The SDN paradigm has been proposed as a way to facilitate and foster Internet evolution by enabling innovation through network programmability. The main idea behind SDN is to decouple the control from the data plane by: (1) removing control decisions from the forwarding hardware, (2) allowing the forwarding hardware to become "programmable" via an open interface, and (3) having a separate entity called "controller" to define by software the behaviour of the network formed by the forwarding infrastructure, thereby creating a "software-defined network". We contend that, based on its knowledge and control of the network infrastructure, the SDN controller will be able to efficiently orchestrate the capacity sharing efforts involving end-user devices as well as network access elements such as access points, base stations, etc. In exploring SDN-enabled capacity sharing in UCNs, we describe our proposed architecture as well as functions such as mobility management, node admission control, fault tolerance, and load balancing. We also briefly discuss extending the original "logically centralized" SDN paradigm so it can operate in distributed, decentralized UCN environments.

Bruno A. A. Nunes, and Thierry Turletti are with INRIA, France, {bruno.astuto-arouche-nunes, thierry.turletti}@inria.fr

Mateus A. S. Santos, Bruno T. de Oliveira and Cintia B. Margi are with University of São Paulo, Brazil, {mateus,btrevizan,cbmargi}@larc.usp.br

Katia Obraczka is with UC Santa Cruz, katia@soe.ucsc.edu

II. Background and Related Work

In this section, we provide a brief overview of User Centric Networks (UCNs) and discuss some user-centric networking initiatives. We then describe the Software-Defined Networking (SDN) paradigm and discuss its potential to enable and foster efficient capacity sharing in UCNs.

A. Capacity Sharing in User-Centric Networks (UCNs)

UCNs typically refer to wireless network deployments where end users share network resources and cooperate to provide network services. According to [2], UCN network sharing models include: direct sharing, multi-hop networks, and user-enabled *micro-providers*.

In direct sharing, cooperation is enabled by the user if and when the user is available/willing to cooperate, for instance, by sharing network connectivity (e.g., opening access to his/her WiFi connection to other users). Resource sharing can also be enabled by the network operator; consider for example the case of network provider A allowing subscribers of network provider B to access A's "hotspots" when they are in their vicinity and network provider B reciprocates and allows A's subscribers to access its "hotspots". In fact, there are currently a number of capacity sharing services that are commercially available in the context of WiFi access networks. Notable examples include FON[1] and Whisher[2], where users receive incentives from their Internet Service Providers (ISPs) to share their WiFi access.

Self-organizing, autonomous, multi-hop wireless networks, or MANETs, have been the focus of a vast body of research since the mid 1990's [3]. The Disruption Tolerant Networking (DTN) paradigm [4] that emerged in the early 2000's is another notable research thrust which addressed the problem of providing communication services in "extreme" and "connectivity-challenged" environments. Unlike the Internet, in these "extreme" environments, continuous end-to-end connectivity cannot be assumed. Originally motivated by interplanetary and deep space communication scenarios, DTNs also find applications in environmental and habitat monitoring, bridging the digital divide, emergency response, disaster relief, law enforcement and special operations. DTNs also attracted considerable attention from the research community; in particular, the HAGGLE project [5], which was proposed in the context of Pocket Switched Networks (PSNs) [6], is one of the early efforts recognising the importance of user- and content-centricity in the context of network environments prone to episodic connectivity. HAGGLE explored the use of "opportunistic" information dissemination mechanisms, where human factors and mobility patterns play an important part.

More recently, motivated by the wide availability of portable computing devices and wireless communication infrastructure and inspired by new user- and content centricity networking paradigms, projects such as SOCIALNETS [7] and BIONETS [8] have been proposed. SOCIALNETS considers the social interactions between users and how those can be exploited for content delivery, focusing on issues of security and trust. The main goal of the BIONETS project is to provide an integrated network and service environment that scales to large numbers of heterogeneous devices. BIONETS' scalability and adaptability are inspired by biological- and social systems, in which large populations are able to reach efficient equilibrium states and develop effective collaboration strategies. Another initiative worth mentioning was the IETF-sponsored Mobile Ad hoc Networking Interoperability And Cooperation (MANIAC) 2013 Challenge in which participants proposed, implemented, and demonstrated strategies for mobile data offloading in MANETs given cooperation incentives.

The concept of micro-providers [2] refers to end-users who not only can act as consumer/producer of content, but also as provider of network access. In this context, we highlight the ULOOP [9] and PERIMETER [10] projects. ULOOP exploits how user-provided network access can help in expanding the coverage of a multi-access backbone infrastructure. Furthermore, ULOOP focuses also on other important aspects such as legislation implications, community-driven services, trust management, cooperation incentives, and how these aspects enable new business models for both users and access providers. The main goal of the PERIMETER project is to set a baseline for future user-centric mobility experimentation focusing on security, Quality of Experience (QoE), and also cooperation and trust in mobile networks.

B. Software-Defined Networking (SDN)

The basic premise of the Software-Defined Networking (SDN) paradigm is to decouple the network control- and data planes to facilitate network protocol and service evolution, especially in production networked environments. In SDN, the network intelligence is logically centralized in software-based "controllers" (the control plane), and network devices become simple packet forwarding devices (the data plane) that can be programmed via an open interface (e.g., ForCES [11], OpenFlow [12], etc), which would enable programmatic control of the network's data plane.

As illustrated in Figure 1, the separation between the forwarding hardware and the control logic allows easier deployment of new protocols and applications, straightforward network visualization and management, and consolidation of various middleboxes into software control. Instead of enforcing policies and running protocols on a convolution of scattered devices, the network is reduced to "simple" forwarding hardware and decision-making network controller(s). A brief history of programmable networks, SDN's current state-of-the-art, as well as current research on SDN can be found in [13].

In [14], we describe our preliminary ideas towards providing capacity sharing services enabled by SDN. In this paper, we go a step further and discuss a general, yet simple SDN-based framework and architecture for network resource sharing in user-centric networking environments. The proposed framework addresses some of the main challenges involved in enabling capacity sharing in UCNs, such as resource discovery, node admission control, support for mobility, cooperation incentives, QoS, and security.

[1] https://corp.fon.com

[2] http://www.whisher.com

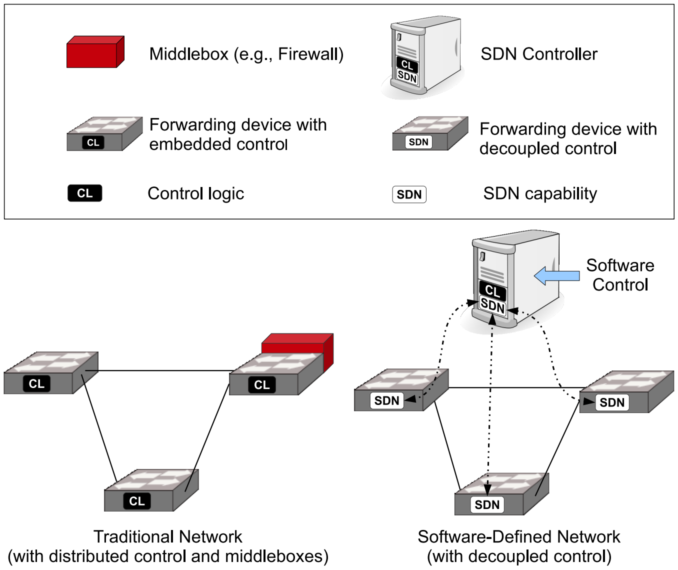
Middlebox (e.g., Firewall)
SDN Controller
CL
SDN
Forwarding device with embedded control
Forwarding device with decoupled control
Control logic
SDN capability
Software Control
Traditional Network (with distributed control and middleboxes)
Software-Defined Network (with decoupled control)

Fig. 1. The SDN architecture decouples control logic from the forwarding hardware and enables the consolidation of middleboxes, simpler policy management, and new functionalities. The solid lines define the data-plane links and the dashed lines the control-plane links.

III. UCN Capacity Sharing Challenges and Requirements

As mobile devices and wireless communication become prevalent, users will demand uninterrupted, high–quality communication services regardless of location or type of network access. Our premise is that self–organising, user–centric networks (UCNs) will provide cost-efficient opportunities to meet user demand for communication services in future internets.

Another major challenge for next-generation internets is the need to support a variety of network access technologies, applications, as well as end-user devices. The latter will likely be highly heterogeneous in terms of battery life, operating systems, communication range, processing and storage capabilities, etc. Moreover, the deployed wireless infrastructure will likely be composed of a number of Radio Access Technologies (RATs) (e.g. WiFi, WiMAX, 3GPP, LTE, etc.) that may be geographically co-located. The motivation behind this is the fact that single RAT systems are not able to offer ubiquitous coverage and QoS guarantees. Each RAT is typically composed by a set of cells, where each cell is covered by a Radio Access Point (RAP), such as WiFi access points, WiMAX routers, or 3G base stations.

User requirements and preferences also play a very important role especially in the context of UCNs. Combined with network policy enforcement and resource availability they will determine when and where new users can access network services. In the remaining of this section, we discuss the many challenges posed when providing capacity sharing services in UCN scenarios and the resulting functional requirements. We also explore how an SDN-enabled capacity sharing architecture can address these challenges and summarize the proposed SDN-based mechanisms in Table I.

A. *Resource Sharing and Allocation*

Achieving efficient resource allocation is a fundamental challenge when providing network capacity sharing services in UCNs. It must account for network resource availability and usage, as well as consider user quality-of-service requirements. The latter is discussed in Section III-C below.

Similar to admission control mechanisms such as Joint Call Admission Control (JCAC) for cellular networks, we propose the Node Admission Control (NAC) mechanism. NAC will determine whether a new end-user (node) can be admitted into the network and if so, with which RAP (and thus RAT) it should be associated. Several previous JCAC algorithms consider user preferences in making RAP (and RAT) selection using for example (1) multiple-objective decision-making (MODM), (2) converting imprecise variables into quantitative values, or (3) adopting a fuzzy multiple-attributes decision-making (MADM) approach. In the context of UCNs the decision or selection of the most suitable RAP should consider, among other things, network usage and resource availability. In an SDN-based capacity sharing architecture, the SDN controller can use its global knowledge of the network topology and conditions to decide whether new users can be admitted and if so, how much resources can be allocated and which RAP will be used. SDN-enabled node admission control (NAC) can be implemented as an application running on the SDN controller. As such, the SDN controller's "NAC module" can decide the best available RAP for a particular user based on the controller's knowledge of the current network topology and conditions. We describe the proposed SDN-enabled NAC mechanism in more detail in Section IV.

In the case of WiFi for example, it is worth noting that, even though IEEE 802.11 specifies that a user should be associated with a single access point, and that the user is the one responsible for selecting its point of attachment, standardization initiatives confirm the demand for moving away from the user-driven association model. Examples include the WiFi Alliance Hotspot 2.0[3], which enables devices to automatically discover and securely connect to WiFi hotspots with no user intervention, and the IETF work on network-based mobility management solutions such as Proxy Mobile IPv6 (PMIPv6) [15] and Mobile IPv6 (MIPv6) [16].

However, interactions between existing standards present compatibility issues [17] and SDN has emerged as a promising solution due to its flexibility, ease of deployment and management. For example, the work of Dely et al. [18] proposes virtual SDN switches integrated into user stations in order to allow multiple access point (AP) associations. Such SDN-based schemes can be deployed to manage handovers and allow nodes move between APs. One possibility is to have a mobility management service running on a wireless station make the decision on switching from one AP to another in a decentralized fashion, a la user-driven association. Another promising solution is the use of a centralized NAC, as we propose here, which can: (1) install flows in the virtual SDN switches of participating user stations to define the most suitable AP to be used, and (2) on the network side, adapt

[3] http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-passpoint.

the wired and wireless backhaul accordingly (e.g., establish new routes for the new selected AP). This method can also be used to provide SDN-based services such as load balancing, fault tolerance, and mobility management.

Furthermore, when deploying any new system, backward compatibility is an important consideration. Our SDN capacity sharing framework for UCNs is no exception: it will accommodate legacy WiFi user devices by allowing them to perform traditional “device-driven” association. We discuss this in more detail in Section IV.

B. *Cooperation*

Social interactions and human interests are the basis for building trust; yet trust is an integral component in many kinds of human interactions, allowing individuals to act under uncertainty. Examples may include, exchanging money for goods and services (e.g. reputation models in auction websites such as eBay), giving access to your property, and choosing between conflicting sources of information (e.g. wiki-pages and blogs on the web). All may utilize some sort of trust model. Trust is also the basis for end-user nodes to rely on other nodes for connectivity [2]. Recent work done in the context of ULOOP [9] and PERIMETER [10] have been focusing on trust, cooperation incentives, and reputation-based schemes for cooperation in the context of UCNs.

It is clear that incentive and trust models are necessary to ensure collaboration between nodes by incentivizing end-users, acting as RAPs, to agree to forward traffic to/from other nodes. Incentive schemes may include monetary compensation, reciprocity in the form of network access credits, etc.

C. *QoS Support*

Clearly, fulfilling end user quality-of-service (QoS) requirements is another fundamental challenge that network capacity sharing mechanisms must address. In particular, it is important to both address the requirements of the end user requesting networking services (e.g., maximum delay or minimum bandwidth requirements) as well as to not deteriorate the service provided by end-user nodes who are willing to serve as RAPs. The latter is key to ensure cooperation on the part of currently connected nodes, cooperation incentives aside.

For instance, it is important to limit the fraction of a node’s total bandwidth as provided by the ISP that will be shared with other nodes. This allocation can be adjusted dynamically based on resource availability and allocation policies and can be enforced by the SDN controller through ingress policies. As an example, current OpenFlow versions already allow QoS policies to be enforced by means of creating virtual ports on the switches and applying priority scheduling mechanisms such as weighted fair queuing (WFQ). QoS policies could also be used by service providers in order to offer differentiated services among users. Examples include scenarios in which customers that share their resources might get incentives to do so by means of higher ingress policies, while customers provided with temporary shared services are subject to lower bandwidth. It also allows the controller to restrict access to certain applications (e.g. deny or limit BitTorrent connections or resource-demanding applications such as video streaming) in order to preserve QoS for nodes serving as RAPs. Here too, the use of an SDN-based architecture allows the SDN controller to implement and enforce QoS policies by employing techniques such as load balancing among RAP nodes, enforcing flow priorities, etc.

D. *Security*

In order to control access to network resources it is required not only to authenticate a new end-user node, but also to ascertain membership eligibility and bootstrap security services such as data confidentiality and authenticity. Clearly, security is a major concern as existing standards (e.g., 802.1x[4]) do not provide adequate security for these types of scenarios and applications. For instance, in the particular capacity sharing scenario of Figure 2, a RAP node may need to authenticate an end-user node requesting communication services in order to make sure it is a legitimate user. Similarly, nodes should not be able to impersonate RAP nodes in order to benefit from incentives. Furthermore, RAP nodes should not be liable for misbehaving users connecting through them. At the same time, data confidentiality and data integrity should be provided to users connecting through other nodes.

E. *Resilience*

Robustness to failures in order to avoid service disruptions is also key when providing capacity sharing based communication services. In case the current RAP node fails or gets disconnected, end-user nodes connecting through it should be migrated in a seamless fashion to other RAPs. When the failed RAP comes back online, load balancing mechanisms will determine whether to migrate users back to the RAP. This seamless migration of users in response to faults can be supported quite naturally through the SDN controller in an SDN-enabled capacity sharing architecture. In fact, fault tolerance and load balancing can use common basic functions such as topology discovery, network measurement collection, and mobility management.

Control plane robustness is another fundamental challenge and can be addressed by physically distributing control. In the context of SDN, controller functions can be replicated in a number of devices which could assume control in case the current controller fails. This of course requires that controller replicas communicate periodically to: keep their state consistency with one another, detect failures, and select a controller to take over in case of failure of the current controller.

As part of our ongoing work, we have also been exploring logically distributing the SDN controller which addresses not only fault tolerance but also administrative decentralization, in particular when considering internets consisting of infrastructure-less, self–organizing networks that are prone to episodic connectivity.

[4] http://www.ieee802.org/1/pages/802.1x-2004.html.

Challenges	Proposed Solutions
Resource sharing	Node admission control (NAC), resource discovery and network measurements
Cooperation	Incentives
QoS support	NAC, load balancing, mobility management and handover
Security	Identity Based Cryptography
Resilience	Control- and data-plane fault tolerance

TABLE I
CHALLENGES AND SOLUTIONS IN ENABLING CAPACITY SHARING IN UCNS.

IV. SDN-ENABLED CAPACITY SHARING ARCHITECTURE FOR UCNS

We contend that SDN facilitates and fosters user-centric capacity and resource sharing services by consolidating in the SDN controller network control functions as well as network structure and topology knowledge. This section provides an overview of our proposed SDN-enabled capacity sharing framework, which we call User-Centric Networking - Capacity Sharing (UCN-CS), and how it addresses the different challenges discussed in Section III.

A. Architecture Overview

As illustrated in Figure 2, the architectural components of the proposed UCN-CS framework include the Network Gateway (NGW) and Requesting Node (RN) which are described as follows:

- **Network Gateway (NGW)**: An SDN-enabled device offering gateway services through which end users can connect to the network infrastructure (e.g., the Internet). It can be an end-user device, where the user is willing to share connectivity, or an SDN-enabled Radio Access Point (RAP) such as WiFi access points, WiMAX routers, etc. NGWs run UCN-CS' NGW services as depicted in Figure 2(b). Note that NGWs rely on an SDN-enabled switch in order to forward traffic accordingly. In terms of their implementation, SDN-enabled switches may, for example, comprise an instance of an OpenFlow client and act as an SDN forwarding device (e.g., using an OpenFlow software-switch such as *Open vSwitch* [19];
- **Requesting Node (RN):** Typically an end-user device which can use an NGW as a provider of connectivity and networking services. RNs run UCN-CS' RN services as illustrated in Figure 2 (c). Note that RNs do not need to be SDN-enabled since they do not forward traffic. For incremental deployment purposes, when communicating with legacy devices, NGWs will fall back and provide compatible connectivity services, e.g., WiFi, WiMAX, etc.

As previously discussed, the cost of provisioning the current network infrastructure by increasing and upgrading radio coverage can be prohibitive. Furthermore, it may also be inefficient in terms of network resource utilization especially in the case of over-provisioning to meet peak demand. Our proposed network capacity sharing architecture broadens the scope of access network infrastructure and provides ubiquitous connectivity in a scalable- and resource-efficient fashion; it does so by relying on an already deployed network of wireless (mobile) end-users. For example, in the case of the scenario depicted in Figure 2, RNs connect to the infrastructure via other end-user SDN-enabled devices acting as NGWs. The RNs can be directly connected to the NGW, connected via multiple NGWs, or reach a NGW via multi-hop routing in a MANET. The NGWs are controlled by an SDN controller and execute forwarding rules at the controller's command.

In the case where RNs are part of a MANET, they can reach NGWs through the MANET routing protocol run by the MANET, or by letting the SDN controller itself define the routes and install them in the MANET nodes. The second case presents many interesting opportunities and benefits, but also many challenges and open research issues. For example, relying on the global view of the network at the SDN controller enables a number of services, such as resilience to link failure, fast route (re-)computation, load balancing, etc. In this context, the trade-offs between these added services and impact of the extra traffic overhead and delay due to SDN control operations remain to be evaluated against routing protocol overhead, performance, and services enabled by legacy MANET routing mechanisms.

MANETs are inherently decentralized and in some cases may be intermittently connected to the network infrastructure due a variety of factors such as wireless channel impairments, power limitations and mobility of the participating nodes, to list a few. Consequently, relying on a centralized SDN controller may not be viable. We argue that a decentralized SDN control plane approach is more adequate in such inherently distributed scenarios. If we consider the MANET example, any SDN-enabled ad-hoc node could assume the role of SDN controller for the MANET when needed (e.g., in the case of network partition). In this case, eligible MANET nodes run an election protocol among them in order to define the most suitable candidate to take over the control of the MANET when needed. Decentralization and distribution of control in challenged networks remains an under-explored field and is one of the targets of our ongoing research efforts.

One distinguishing feature of the proposed framework is that security is an integral part of its capacity sharing services. For instance, in order to control access to network resources, a new RN must be authenticated beforehand and have its subscription verified; data confidentiality and authenticity can also be offered. Currently the basic operations provided by our framework[5], which are illustrated in Figure 3, are as follows:

- **Gateway discovery:** NGW nodes, via their UCN-CS layer (see Figure 2(b)), send periodic messages announcing their gateway services. Upon receiving these messages, an RN will choose an NGW by sending a `Request` message enabled by the RN's UCN-CS layer (see Figure 2(c)) to the selected NGWs. Such requests will be forwarded to the SDN controller, which will then choose the best NGW (e.g., based on the NAC's output) and assign it to the RN;

[5] Such functionalities have already been implemented and are available for download at http://inrg.cse.ucsc.edu/community/Software

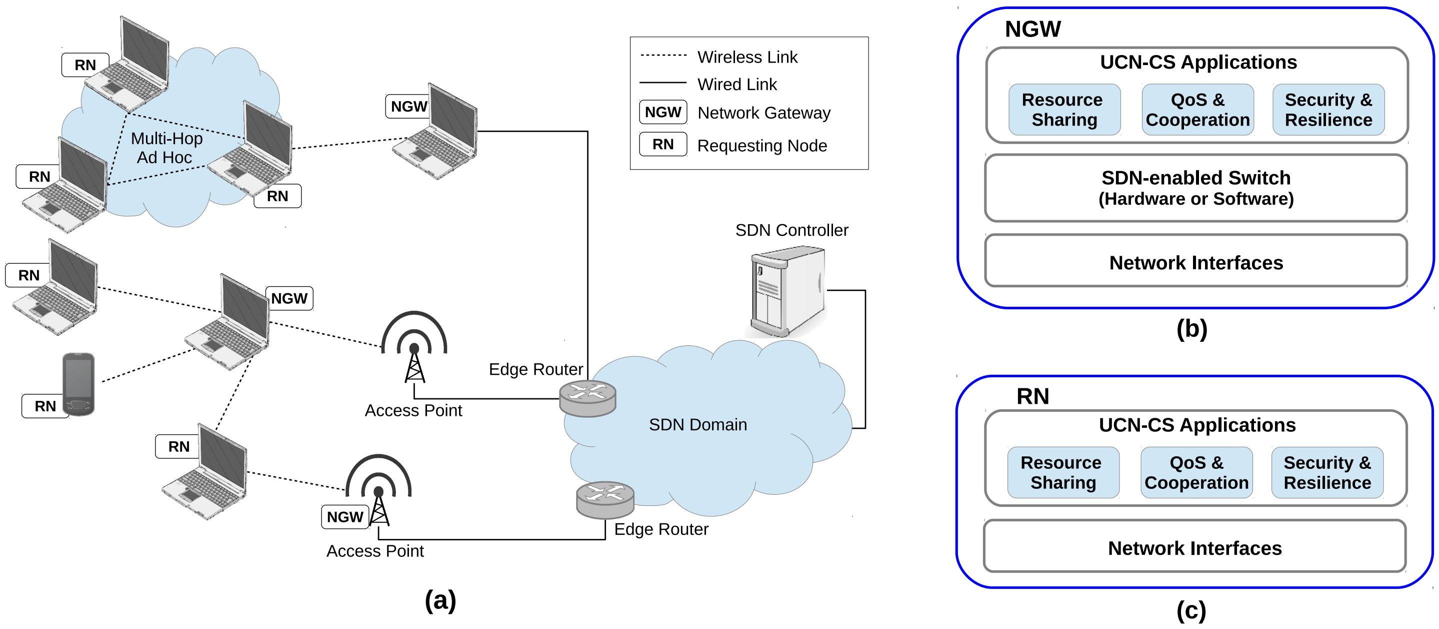

Fig. 2. Capacity sharing architecture overview (a), Network Gateway (NGW) and Requesting Node (RN) architectures (b and c, respectively).

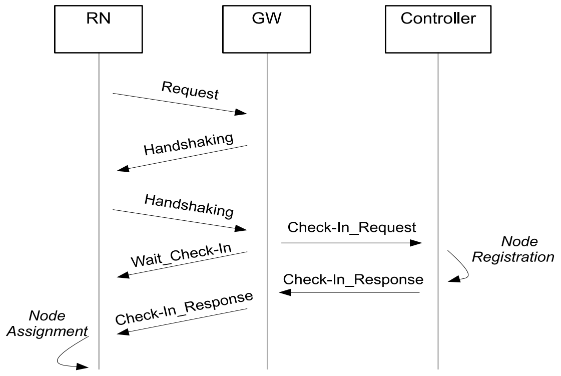

Fig. 3. Basic capacity sharing operations: gateway discovery, handshaking, RN check-in.

- **Handshaking**: an NGW node chosen by the NAC mechanism responds to a user request and initiates a handshaking procedure for node authentication;
- **RN check-in**: the NGW requests authorization from the SDN controller, which queries its database in order to approve allocation of resources to the designated RN.

When an RN is authorized, the SDN controller adds the proper new entries in the flow table of the selected NGW as well as the flow table(s) in the forwarding devices on the RN's data path towards the Internet.

B. Node Admission Control

In order to provide efficient resource sharing and utilization, we propose a Node Admission Control (NAC) mechanism as illustrated in Figure 4. A NAC works as a service running on the SDN controller and uses input from the user– and forwarding devices. More specifically, the SDN-enabled NAC will receive information from SDN-capable devices (e.g., RAP devices that are able to communicate with the SDN controller and are capable of implementing and executing forwarding actions and rules) and the SDN controller's knowledge base, that may include policy rules and topology information. The NAC's decision engine would then be able to make decisions about whether to admit a new node, migrate nodes among RAPs, etc. NAC's decision engine "communicates" with the OpenFlow engine, which then sets the appropriate forwarding rules at the new elected RAP device and potential forwarding devices in the data-path between the new end-user device and the Internet. At this point, topology information should be also updated to maintain consistency.

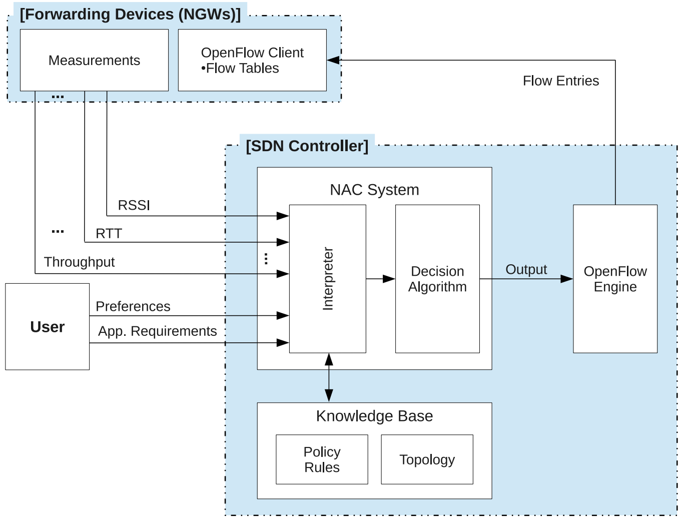

Fig. 4. Node Admission Control (NAC).

The trade-offs between the amount of overhead incurred (e.g., information exchanged, stored, and processed) and the resulting accuracy and responsiveness need to be considered when designing a NAC. There are interesting research opportunities in addressing these trade-offs in order to design efficient

NAC.

C. Resource Discovery and Network Measurements

Future internetworks will likely grow increasingly heterogeneous in terms of the devices they interconnect and the networks and links used to interconnect them. Therefore, a variety of factors should be considered when choosing an end-device as an NGW. Such factors range from battery lifetime to network connectivity and trust. An SDN controller that collects this information periodically is able to make informed decisions about when and where to admit new nodes (i.e., performing NAC, as illustrated in Figure 4), or to which RAP to handover already connected RNs (e.g., performing load balancing).

Information about current network conditions is key to support decisions such as admission control, quality-of-service and mobility management. In the case of current SDN standards like OpenFlow, basic network measurements such as port- and queue statistics can be used to estimate link bandwidth availability of RAP devices. Network statistics can be collected by NGWs and provided to controllers, which in turn can then use available bandwidth and queue statistics to decide when and where to handover RNs. For instance, a RAP experiencing high load can have some of its RNs offloaded to another gateway device, which will likely benefit the RNs as well since they are likely to experience better service.

SDN-enabled devices can be queried periodically by the controller in order to collect relevant statistics, as depicted in Figure 4. This figure shows a few examples of metrics/statistics, relevant to the NAC's decision making process, including Received Signal Strength (RSSI) and Round Trip Time (RTT). However, it is worth noting that in heterogeneous network technologies, link quality can be assessed differently, and different measurement mechanisms could be considered. For instance, *Open vSwitch* provides an interface for measuring bandwidth. However, delay related statistics cannot be obtained using current standard SDN implementations.

D. Security

In order to provide security services such as confidentiality, integrity, and mutual authentication, several cryptographic schemes could be used. It is important then to discuss some of the trade-offs related to choosing a particular cryptographic method, addressing efficiency and increasing the resilience to attacks such as impersonation, unauthorised data access, or data modification.

An SDN domain is composed of an SDN controller, SDN-capable devices, that may be acting as NGWs, and general end-user devices as RNs. We argue that Identity Based Cryptography (IBC) [20] is well suited to provide simple yet efficient security services in an administrative domain, which is the case of ISPs and their customers and companies and their employees.

IBC allows a user to calculate a public key from an arbitrary string. Choosing the user's identity as a public key has advantages such as: (1) there is no need to verify the public key using an online Certification Authority (CA); and (2) a user only needs the recipients' identities in order to calculate public keys (i.e., there is no need to ask for public keys). Thus, IBC-based cryptographic protocols are simple and efficient since they eliminate the need for generating and managing users' certificates. A user's public key is used as the user's identity, and the question then becomes how to obtain the corresponding secret key.

In IBC schemes, a Trusted Third Party (TTP) is responsible for secret key generation, which is performed by using the TTP's secret key, also known as master secret key, and the public key of the target user. Note that all secret keys can be computed by the TTP. Fortunately, in the scenario explored here, there is a synergy present between SDN controllers and TTPs. Controllers can be regarded as trusted entities, since they provide interfaces to applications that perform management tasks. Thus, in the context of IBC, a controller could be responsible for generating (and possibly distributing) private keys to users in its domain.

However, when inter-domain functions and services are considered (e.g., cooperation between different ISPs), IBC may not be adequate. Since the TTP can impersonate all users on the network and access and modify transmitted data, it is strongly recommended that the TTP be managed by a single owner. In this case, public key distribution schemes need to be implemented. Additionally, public key validation will be needed to ensure that a public key received from the network is indeed the correct one. This is usually implemented using a Public Key Infrastructure (PKI), which then requires a trusted entity, known as CA, to be online to respond to certification requests (i.e., a certificate for the requested public key and identity).

E. Mobility Management and Handover

With the proliferation of wireless mobile devices connecting to the Internet, there is a need for efficient and scalable mobility management in order to guarantee uninterrupted network services while maintaining desired levels of QoS. In this context, SDN enables fast and transparent switching of RNs between NGWs. This can be achieved by instantiating an SDN software switch in the RN, which will then, be able to operate as a bridge with virtual interfaces, each one associated with a single NGW [21]. The "best" NGW could be used as a primary access device to the infrastructure network.

In a densely deployed access network, the choice of the NGW can be made by the user device or the network infrastructure. This also defines which one will be in charge of mobility management. Should the RNs be in charge, they need to be provided with information for deciding where and when to switch (e.g. signal strength, link quality statistics, etc). On the other hand, the SDN controller's global network knowledge allows it to provide centralized NAC, in which users are associated with NGWs automatically, based on users preferences and availability of resources. Alternatively, the controller could simply only expose to each RN the particular NGW with which it wants the RN to associate [22].

F. Incentives and Trust Models

In an SDN-enabled capacity sharing framework, the SDN controller can serve as the entity that will provide incentives to end users to share their connectivity with others. The SDN controller will also be responsible for authenticating nodes that are willing to serve as RAPs as well as end-user nodes. Unlike the current SDN paradigm which relies on (logically) centralized control, in an distributed SDN model, the "main" SDN controller for a particular domain (e.g., ISP) could delegate certain decisions (e.g., whether to agree to forward traffic for a particular node or set of nodes) to "local" controllers (e.g., RAPs).

Furthermore, it is not enough that control messages successfully and securely reach their destination; both endpoints must be able to trust each other to act properly. Forwarding nodes need to be able to trust that the discovered controller is not malicious before accepting control. Likewise, the controller must be able to trust that forwarding nodes, that have accepted control acting as NGWs, are correctly following instructions. For this trust to exist, mechanisms must be in place to ensure the legitimacy of nodes/controllers, the authenticity of the control traffic, and verify that devices act as expected in response to instructions. Additionally, with a global view of the network, an SDN controller can decide whether to delegate forwarding capabilities to potential forwarding nodes or even to permit or prevent access from RNs and other devices based on an eligibility function implementing needed trust and reputation models.

G. Load Balancing

Detecting overloaded NGWs is important to guarantee adequate network performance. This can be achieved by using centralized NAC to switch a RN to a new NGW. As previously pointed out, the SDN controller with its ability to obtain global knowledge of the network can decide to migrate RNs among NGWs in order to balance network load and thus offer adequate service to end-users. An interesting challenge to consider is how to prevent possible oscillation (or "ping-ponging") of a RN between NGWs. Since the network is densely deployed, redundancy can be used to avoid excessive user device migration situations. For example, an already migrated RN that further experiences low throughput can be provided with services by more than one NGW simultaneously. Moreover, when moving RNs from one NGW to another, based on available resources and load conditions, the NAC system may be in charge of such decisions, and a hysteresis approach must be taken in order to efficiently reduce both the handover initiation delay, avoid ping-pong behaviour (i.e. multiple and consecutive associations and disassociations between a group of NGWs), and decrease the number of unnecessary handovers.

H. Fault Tolerance

The goal of fault tolerance at the data plane is to detect link failures and take recovery actions. In the case of Openflow, there is no topology monitoring specifications, which can be implemented by leveraging Link-Layer Discovery Protocol (LLDP) messages or by customizing switch functionalities [23].

In the context of UCNs, node failures should also be considered, which is somewhat related to load balancing. Keeping track of nodes not only allows to detect an overload situation, but also to detect a node failure. In the latter case, SDNs can be useful for acquiring the measurements from the forwarding devices, as discussed previously, and to set up the flows from the Internet to a new NGW, via alternative paths, as needed. A centralized NAC system can maintain a location database, so that the flow affected by a NGW failure would be redirected accordingly.

Controller faults can be dealt with by a variety of methods. SDN implementations such as OpenFlow consider a fail-safe mode for SDN-enabled devices, so that packets can be forwarded by using the same method as traditional L2 switches. A rather efficient design choice is to distribute the control plane, using a single controller in charge while the others operate as replicas [24].

As previously pointed out, we are also exploring the idea of logically distributing the control plane using a hierarchy of controllers. According to the control hierarchy, controllers at different hierarchical levels will be responsible for different control functions. The control hierarchy tries to match the internet's hierarchical structure where the "main" controller resides at the backbone level, and "secondary" controllers are responsible for regionals, stubs, etc. This decentralized control model is also well-suited for internets consisting of infrastructure-based as well as self-organizing networks which may be frequently disconnected from the infrastructure.

V. Concluding Remarks

In this article, we explored Software-Defined Networking (SDN) as a promising approach to address the challenges raised by User Centric Networks (UCNs), in particular providing efficient network capacity sharing services. We contend that, based on its knowledge and control of the network infrastructure, the SDN controller will be able to efficiently orchestrate capacity sharing efforts involving end-user devices as well as network access elements such as access points, base stations, etc. In exploring SDN-enabled capacity sharing in UCNs, we discussed requirements and challenges raised by sharing network resources in a scalable and efficient manner, and proposed a simple, yet general framework that includes functions such as mobility management, node admission control, fault tolerance, and load balancing. One distinguishing feature of our SDN-based capacity sharing approach is that security is an integral part of the proposed framework.

Acknowledgments

The authors would like to thank the reviewers for their careful examination of the manuscript and valuable comments which helped to considerably improve the quality of the paper. This work is partly funded by the Community Associated Team between INRIA and UCSC and the French ANR under the "ANR-13-INFR-013" project, and by NSF grant CNS

1150704. It was also partly funded by Brazil CNPq under the grant 245588/2012-4, and So Paulo Research Foundation (FAPESP), Brazil, under the grant 2013/15417-4.

References

[1] Marc Mendonca, Bruno Astuto A. Nunes, Katia Obraczka, and Thierry Turletti. Software defined networking for heterogeneous networks. *IEEE COMSOC MMTC E-Letter*, 2013.

[2] R. Sofia and P. Mendes. User-provided networks: consumer as provider. *Communications Magazine, IEEE*, 46(12):86–91, December 2008.

[3] Imrich Chlamtac, Marco Conti, and Jennifer J.-N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13 – 64, 2003.

[4] Zhensheng Zhang and Qian Zhang. Delay/disruption tolerant mobile ad hoc networks: latest developments. *Wireless Communications and Mobile Computing*, 7(10):1219–1232, 2007.

[5] European Commission IST Program. HAGGLE - An innovative Paradigm for Autonomic Opportunistic Communication. 2006 to 2010.

[6] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 244–251, New York, NY, USA, 2005. ACM.

[7] European Commission IST Program. SOCIALNETS - Social networking for pervasive adaptation. 2008 to 2011.

[8] European Commission IST Program FP6-027748. BIOlogically inspired NETwork and Services - BIONETS. 2006 to 2010.

[9] European Commission IST Program FP7 257418. ULOOP project. 2014.

[10] European Commission IST Program FP7 224024. PERIMETER - User-Centric Paradigm for Seamless Mobility in Future Internet. 2008 to 2011.

[11] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810 (Proposed Standard), March 2010.

[12] 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*, 38(2):69–74, 2008.

[13] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti. A survey of software-defined networking: Past, present, and future of programmable networks, 2014.

[14] M.A.S. Santos, B.T. de Oliveira, C.B. Margi, B.A.A. Nunes, T. Turletti, and K. Obraczka. Software-defined networking based capacity sharing in hybrid networks. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, pages 1–6, Oct 2013.

[15] K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6. RFC 5213, RFC Editor, August 2008.

[16] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775, RFC Editor, June 2004.

[17] G. Giaretta. Interactions between Proxy Mobile IPv6 (PMIPv6) and Mobile IPv6 (MIPv6): Scenarios and Related Issues. RFC 6612, RFC Editor, May 2012.

[18] Peter Dely, Andreas Kassler, Lawrence Chow, Nicholas Bambos, Nico Bayer, Hans Einsiedler, and Christoph Peylo. Best-ap: Non-intrusive estimation of available bandwidth and its application for dynamic access point selection. *Computer Communications*, 39(0):78 – 91, 2014. Research advances and standardization activities in WLANs Research advances and standardization activities in WLANs.

[19] Ben Pfaff, Justin Pettit, Keith Amidon, Martin Casado, Teemu Koponen, and Scott Shenker. Extending networking into the virtualization layer. In *ACM SIGCOMM Workshops)*, 2009.

[20] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer Berlin Heidelberg, 2001.

[21] Peter Dely, Andreas Kassler, Lawrence Chow, Nicholas Bambos, Nico Bayer, Hans Einsiedler, and Christoph Peylo. Best-ap: Non-intrusive estimation of available bandwidth and its application for dynamic access point selection. *Computer Communications*, 2013.

[22] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, and Brian Zill. Designing high performance enterprise wi-fi networks. In *NSDI*, volume 8, pages 73–88, 2008.

[23] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takacs, and P. Skoldstrom. Scalable fault management for openflow. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6606–6610, June 2012.

[24] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. B4: Experience with a globally-deployed software defined wan. *SIGCOMM Comput. Commun. Rev.*, 43(4):3–14, August 2013.