Bits and Coins: Supporting Collaborative Consumption of Mobile Internet

Dimitris Syrivelis*, George Iosifidis†, Dimosthenis Delimpasis*, Konstantinos Chounos*, Thanasis Korakis*, Leandros Tassiulas†

*Center for Technology and Research Hellas, Greece

†Dept. of Electrical Engineering, and Yale Institute for Network Science, Yale University, USA

***Abstract*—The recent mobile data explosion has increased the interest for mobile user-provided networks (MUPNs), where users share their Internet access by exploiting the diversity in their needs and resource availability. Although promising, MUPNs raise unique challenges. Namely, the success of such services relies on user participation which in turn can be achieved on the basis of a fair and efficient resource (i.e., Internet access and battery energy) exchange policy. The latter should be devised and imposed in a very fast time scale, based on near real-time feedback from mobile users regarding their needs, resources, and network conditions that are rapidly changing. To address these challenges we design and implement a novel cloud-controlled MUPN system, that employs software defined networking support on mobile terminals, to dynamically apply data forwarding policies with adaptive flow-control. We devise these policies by solving a coalitional game that is played among the users. We prove that the game has a non-empty core and hence the solution, which determines the servicing policy, incentivizes the users to participate. Finally, we evaluate the performance of the service in a prototype, where we investigate its performance limits, quantify the implementation overheads, and justify our architecture design choices.**

I. Introduction

1) Motivation: The last few years we are witnessing an ever increasing demand for mobile data [1] that places unprecedented strain on cellular networks, and forces network operators to adopt expensive data plans [2]. These developments gradually made both the operators and the users keen to explore alternative network access services which can provide low cost Internet connectivity to the users and/or reduce the networks' congestion. In this context, user-provided networks (UPNs) [3] where users offer to each other (mobile) Internet access services, constitute a very promising solution.

This Internet sharing model is not actually new, as it dates back to FON-like WiFi sharing communities [4]. However, UPNs are currently attracting renewed interest with focus this time on handheld mobile devices that can operate, on the spot and on demand, as mobile Wi-Fi access points (APs). Interestingly, a number of innovative companies have already launched such multi-hop/path mobile UPN services [5], [6], [7]. The underlying concept is to exploit the diversity of users needs and resources, as Internet access and battery energy, and create a network effect where the service benefits increase with the number of participants. These models can remedy several problems that have so far dogged users and network operators. In particular:

- *Unsatisfied Demand*. Mobile data plans are expensive [2], and many users consume less than their monthly quota [8]. Hence, users may have unused capacity or unsatisfied needs in each month.
- *Poor Connectivity*. Cellular networks are frequently congested or have poor coverage, offering high-latency and/or low throughput connectivity to disappointed subscribers [9].
- *Unused Capacity*. There is abundance of Internet capacity residing idle at the network edge, either as unused data plans, or at the underutilized WiFi APs [10].

MUPNs can address these issues through on-the-fly (grass-roots) mesh networks that allow the collaboration of the users for sharing the idle capacity at the edge of cellular and WiFi networks, and finding the best Internet access routes. The main idea is presented in Fig. 1. In such architectures, users may act as hosts (gateways or relays), or clients, and exchange roles based on their needs and resource availability, as dictated by the policy of the UPN.

The Challenges. However, devising and imposing routing and flow control policies (servicing policy) for such multi-hop systems, while ensuring global network consistency, is a very intricate task, especially since it is required to recalculate and update these policies in a small time scale. The latter is necessary so as to identify changes regarding the wireless medium conditions, and the needs of the users. Clearly, today's mostly OSI layer-7 cooperative networking schemes (single-hop that rely on tethering) suffer from poor performance and cannot address these issues.

At the same time, such a service needs to ensure that all the users, acting either as service hosts or clients, are willing to collaborate with each other. This incentive provisioning problem should take into account both the performance of each user, in terms of amount of served data, and the incurred costs, in terms of monetary and energy consumption costs. Each user should be ensured that he cannot achieve better overall performance if acting independently (not participating in the service), or if he cooperates only with a subset of other users (colluding). This is of paramount importance so as to avoid strategic user decisions that may impact the efficiency of the service. These requirements further compound the already challenging design and implementation of MUPN services.

2) Methodology and Contributions: Our goal is to propose an efficient architecture for this problem and evaluate a proof-of-concept implementation, based on the latest systems design trends, and on a sound game-theoretic model for the servicing policy. We consider a system where every user (device) is

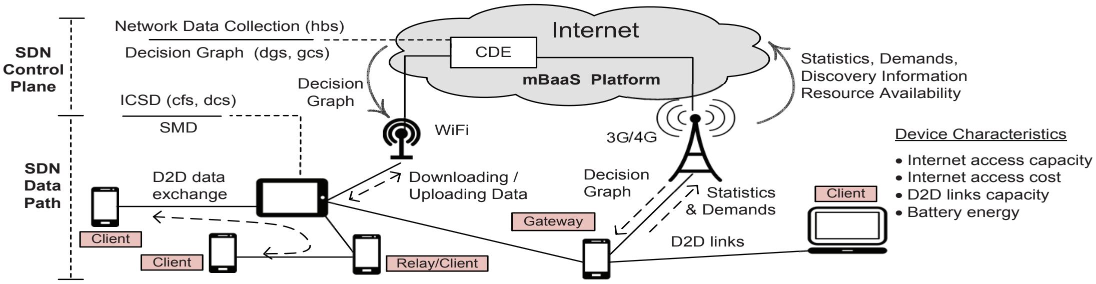


Fig. 1. The collaborative network sharing (CoNeS) MUPN platform: users exchange roles of gateways, clients and relays, and connect to the Internet through multi-hop paths. Each device features a SMart Datapath(SMD) that implements forwarding and runs the Internet Connection Sharing Daemon (ICSD), which is responsible for neighborhood device discovery, link capacity measurements, the user traffic control functions, and the low-level SDN configurations. This information is sent to the backend cloud service called Connection Decision Engine (CDE), which responds with the servicing policy. The devices update accordingly their SDN datapaths in a consistent fashion.

described by his Internet access capacity and cost, device battery energy, and the capacity of links with his neighbors. The users may want to upload or download content to/from the Internet, or directly exchange content (D2D). A high-level overview of the proposed *Collaborative Network Sharing platform* (CoNeS) is shown in Figure 1.

Theoretical Framework. We model the users' interactions as a coalitional game with transferable utilities (TU) [11], and we find an equilibrium solution that satisfies all the participants. This solution determines how much Internet access, relaying bandwidth and battery energy each user should contribute, and how much of the MUPN total capacity (in terms of mobile data) he will receive. Moreover, the solution dictates the side payments among the users that can be realized through a virtual currency system. This system enables users to pay with *coins* for the services they receive, and get paid accordingly, when they serve as gateways or relays, other users (*bits for coins*). We prove that this TU game has a non-empty core, thus the obtained equilibrium policies for the grand coalition (i.e., when all users cooperate) ensure higher payoffs for the participants, compared to all possible deviation scenarios. Clearly, such a solution framework is applicable only if it is fed with up-to-date information.

System Architecture. In order to satisfy this requirement, we design a state-of-the art mobile Backend as a Service (mBaaS) platform that runs in the cloud, and leverages the Software Defined Networking (SDN) paradigm to coordinate in very fast time scale with each user's device. Every device runs an SDN-based light-weight client service (ICSD) which performs all the necessary local tasks, such as network discovery, statistics collection and device profiling (e.g. battery level), and communicates it periodically to the cloud service (CDE). The latter monitors the system through these *hearbeat* messages, and finds the solution of the TU-game that dictates the servicing policy (or, *decision graph*): the assignment of gateway, relay and client roles, the exact flow rate and routing decisions per node, and the coin transfers. The decision graph is subsequently forwarded to the mobile devices which update their SDN datapaths in a consistent fashion.

Implementation and Evaluation. In order to shed light on the very details of the system design, assess its performance, and quantify the actual overheads, we implemented a CoNeS prototype and performed experiments in our testbed, using an embedded platform. These devices, that resemble high-end tablets, were set up with OpenVSwitch (OVS) OpenFlow datapath implementation, and the Linux HTB queuing discipline (for flow rate control). We explored how adaptive such a system can be, i.e., how fast the CDE can coordinate with the ICSDs, and what is the impact of this communication on the service performance and battery energy of the nodes.

To this end, the main technical contributions of this work can be classified as follows:

CoNeS Architecture. We designed a MUPN system, with many mobile (or even fixed) users and multi-hop servicing policies, following the state of the art mobile networking systems design principles (SDN, mBaaS). The proposed architecture significantly departs from similar solutions as it is very flexible, adaptive, scalable, and can be directly extended to include operator controls, e.g., for accounting.

Robust Solution. The service policy and payments are derived from the solution of a coalitional game which we prove it has a non-empty core. This ensures that users are deterred to act strategically by deviating and not applying the decision graphs. This property is of paramount importance for such systems that rely on voluntarily user participation.

Implementation and Evaluation. We implemented a CoNeS prototype in our testbed and executed extensive experiments measuring the performance and the limitations, such as the devices' energy consumption which is critical for MUPNs. We found that the CoNeS coordinating operations, which are necessary to sustain an efficient and adaptive MUPN, have minimal resource consumption overheads and no significant impact on user experience.

The rest of the paper is organized as follows. In Sec. II we introduce the system model and the game-theoretic framework for devising the servicing and payment policies. Sec. III describes the MUPN architecture. In Sec. IV we include implementation details, explain the experimental setup, and present the findings of our evaluation. We discuss related works in Sec. V and conclude in Sec. VI.

II. Resource Allocation Framework

We consider a set $\mathcal{N} = \{1, 2, \dots, N\}$ of mobile users and study the system for a certain time period of the order of several seconds. The users form a mesh network which is modeled by a directed graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{E}$ denotes the set of communication links. For each user $i \in \mathcal{N}$, we define the set of incoming and outgoing one-hop neighbors as $In(i) = \big(j : (j,i) \in \mathcal{E}\big)$ and $Out(i) = \big(j : (i,j) \in \mathcal{E}\big)$, respectively, and denote $\mathcal{N}(i) = In(i) \cup Out(i)$.

The users may download or upload content from/to the Internet, or communicate directly with each other, e.g., for exchanging content. We define the respective sets of users as $\mathcal{N}_d$, $\mathcal{N}_u$, and $\mathcal{N}_c$, which are subsets of $\mathcal{N}$. Each user performs only one such task at a time, hence there exist $|\mathcal{N}|$ data flow commodities in the system. For each user $i \in \mathcal{N}$ we define the value $r_i \geq 0$ of his commodity flow, which denotes the amount of data (in bytes) that he uploads, downloads or exchanges with another user during the current time period. Finally, we introduce the utility function $U_i(r_i)$ for each user i, which is a positive, increasing, continuously differentiable, and strictly concave function, and characterizes his satisfaction. In every period, each user can serve one or more roles as follows: he can be a client node, a relay node, or a gateway node by downloading (uploading) data from (to) the Internet.

Capacities and Flows. Let $C_{ij} \geq 0$ denote the amount of data (in bytes) that can be transferred over link $(i,j) \in \mathcal{E}$ during the period of interest. Let $C_{di} \geq 0$ be the maximum amount of data that user $i \in \mathcal{N}$ can download as a gateway from the Internet, and $C_{ui} \geq 0$ be the maximum amount of data that he can upload to the Internet. In case the medium does not separate upstream and downstream flows, as in WiFi, we assume that the user has an Internet access capacity of $C_i \geq 0$ bytes that is shared among all flows. Each user i pays a price $p_i \geq 0$ for each byte that he downloads or uploads from the Internet. These prices depend on the type of his Internet connection (WiFi or cellular), and on his specific *data plan*. On the other hand, we assume that the D2D data transfers are realized over the unlicensed ISM with no cost.

We denote with $y_i^{(n)} \geq 0$ the data that user i downloads or uploads to the Internet for commodity (n), i.e., for each user $n \in \mathcal{N} \setminus \mathcal{N}_c$. We also define the data amount $x_{ij}^{(n)} \geq 0$ of commodity (n), $n \in \mathcal{N}$, that user i delivers to his neighbor j. The operation of the system is described by the servicing policy which comprises the *Internet access* matrix $\boldsymbol{y} = (y_i^{(n)} \geq 0 : i \in \mathcal{N}, n \in \mathcal{N} \setminus \mathcal{N}_c)$, and the *routing* matrix $\boldsymbol{x} = (x_{ij}^{(n)} \geq 0 : (i,j) \in \mathcal{E}, n \in \mathcal{N})$. Clearly, it should hold:

$$r_i = y_i^{(i)} + \sum_{j \in In(i)} x_{ji}^{(i)},\ \forall\, i \in \mathcal{N}_d, \tag{1}$$

and similarly for each user $i \in \mathcal{N}_u$. The routing and Internet access decisions should satisfy the flow balance equations:

$$\sum_{j \in In(i)} x_{ji}^{(n)} + y_i^{(n)} = \sum_{j \in Out(i)} x_{ij}^{(n)},\ \forall\, i, n \in \mathcal{N},\, i \neq n, \tag{2}$$

while it should hold $x_{ij}^{(i)} = 0$, $\forall\, i,\, j \in Out(i)$.

Each link (i,j) cannot support more data than its maximum capacity. Additionally, since the nodes are in proximity, their transmissions may interfere. This severely impacts the amount of data that can be actually delivered over each link. In order to capture these limitations, we follow the interference models that have been extensively used for backbone mesh networks, e.g. see [13] and references therein. This ensures that the servicing policy satisfy the necessary conditions so as to be implementable.

In particular, for all links $(i,j) \in \mathcal{E}$, it should hold:

$$\sum_{k \in In(i)} \frac{\sum_n x_{ki}^{(n)}}{C_{ki}} + \sum_{m \in Out(i)} \frac{\sum_n x_{im}^{(n)}}{C_{im}} + \tag{3}$$
$$+ \sum_{k \in In(j)} \frac{\sum_n x_{kj}^{(n)}}{C_{kj}} + \sum_{m \in Out(j)} \frac{\sum_n x_{jm}^{(n)}}{C_{jm}} + \frac{\sum_{n \in \mathcal{N}} y_i^{(n)}}{C_i} \leq 1$$

Note that we include $y_i^{(n)}$ in (3) only if the respective Internet connection is realized over Wi-Fi links[1]. Moreover, each node $i \in \mathcal{N}$ cannot use more than his available network interface cards $\rho_i \in \mathcal{Z}^+$, including his cellular interfaces, hence:

$$\sum_{n \in \mathcal{N}} \frac{y_i^{(n)}}{C_i} + \sum_{k \in In(i)} \frac{\sum_n x_{ki}^{(n)}}{C_{ki}} + \sum_{m \in Out(i)} \frac{\sum_n x_{im}^{(n)}}{C_{im}} \leq \rho_i\,. \tag{4}$$

Energy Consumption. Each user $i \in \mathcal{N}$ that participates in this service perceives a different amount of dissatisfaction due to the consumption of his battery energy. We use the energy cost function $T_i(e_i)$ to capture this effect, which is strictly convex and increasing in the total amount of consumed energy e_i (in joules).

Let $e_{ij}^s > 0$ be the energy that user i consumes when he sends one byte to user $j \in Out(i)$. Also, $e_{ij}^r > 0$ is the energy that user j consumes for receiving one byte from user i. Finally, $e_i^s > 0$ and $e_i^r > 0$ are the energy consumptions when node i uploads or downloads, respectively, one byte from/to the Internet. Typically, energy consumption is higher with cellular than with WiFi connections and depends on the transmission and reception rates [14]. The aggregate consumed energy e_i is:

$$e_i = \sum_{j \in Out(i)} e_{ij}^s \sum_{n \in \mathcal{N}} x_{ij}^{(n)} + \sum_{j \in In(i)} e_{ji}^r \sum_{n \in \mathcal{N}} x_{ji}^{(n)}$$
$$+ e_i^s \sum_{n \in \mathcal{N}_u} y_i^{(n)} + e_i^r \sum_{n \in \mathcal{N}_d} y_i^{(n)},\ \ \forall\, i \in \mathcal{N} \tag{5}$$

Virtual Coins. We assume that there is in place a virtual currency system that allows users to pay each other for the services they receive and offer. This is very important as it incentivizes the users to participate in the service, even if they do not have communication needs. Such a system can be easily implemented centrally with the proposed mBaaS architecture. In particular, we denote with $z_i \in \mathbb{R}$ the amount of coins that user $i \in \mathcal{N}$ receives ($z_i > 0$) or pays ($z_i < 0$) when he participates in the service. The exact payments, z_i, $\forall i \in \mathcal{N}$, will be determined by the solution of a specific optimization problem, as we explain in the sequel.

User Payoff. Let $V_i(\cdot)$ denote the payoff that user $i \in \mathcal{N}$ receives for each period that he participates in the mobile UPN.

[1]We do not consider interference among different cellular connections as these are managed by the respective cellular base stations. Also, we focus hereafter on Wi-Fi links. The analysis is similar for cellular links using C_{ui} and C_{di} instead of C_i for the sets $\mathcal{N}_u$ and $\mathcal{N}_d$, respectively.

This includes the energy consumption cost, and the monetary cost for the mobile data usage. Hence, it is:

$$V_i(\boldsymbol{x}_i, \boldsymbol{y}_i) = U_i(r_i) - T_i(e_i) - p_i \sum_{n \in \mathcal{N}_u \cup \mathcal{N}_d} y_i^{(n)}. \quad (6)$$

Obviously, the payoff $V_i(\cdot)$ decreases monotonically with the amount of data that user i downloads, uploads or routes. Hence, the user will have no incentive to perform these tasks unless he is compensated accordingly. On the other hand, when user i does not participate in the service, he determines independently the value of $y_i^{(i)}$ which should be balanced so as to increase the user's utility but not induce very high data usage cost. We denote with V_i^s the achieved *standalone performance* according to this optimal decision of each user $i \in \mathcal{N}$.

MUPN Coalitional Game. In this context, the question that arises is the following: *How much data should be served for each user, and how many coins should he receive or pay for the services he offered or received from the UPN*? To address this question, we model the mobile UPN operation as a coalitional game with transferable utilities [11].

Definition 1: The UPN coalitional game is defined as $\langle v, \mathcal{N} \rangle$, where $\mathcal{N}$ is the set of cooperating users, and $v(\cdot)$ is the *characteristic* function which associates to each subset of cooperating users $\mathcal{S} \subseteq \mathcal{N}$ (coalition), a real number $v(\mathcal{S})$ that represents the maximum aggregate payoff increase that coalition $\mathcal{S}$ can achieve and share in an arbitrary way (through coin transfers).

The key issue in TU coalitional games is whether all the users $\mathcal{N}$ will agree to cooperate, forming the *grand coalition*, or if there will be formed separate subgroups. Formally, this question translates to whether the *core* of the UPN coalitional game is empty or not.

Definition 2: For any real valued vector $\boldsymbol{\phi} = (\phi_i \geq 0 : i \in \mathcal{N})$ and any coalition $\mathcal{S}$, we let $\phi(\mathcal{S}) = \sum_{i \in \mathcal{S}} \phi_i$. This vector is called an imputation if $\phi(\mathcal{N}) = v(\mathcal{N})$, and $\phi_i \geq v(\{i\})$ for all $i \in \mathcal{N}$. The core of the game is the set of all imputations $\boldsymbol{\phi}$ for which it holds:

$$\mathcal{I} = \{\boldsymbol{\phi} : \phi(\mathcal{N}) = v(\mathcal{N}), \phi(\mathcal{S}) \geq v(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{N}\} \quad (7)$$

Interestingly, the UPN coalitional game has a non-empty core. Let us first introduce the MUPN optimization problem (MOP) that yields the total welfare increase, i.e., the value of the characteristic function, for the set of $\mathcal{N}$ cooperating users:

$$\max_{\boldsymbol{x}, \boldsymbol{y}} \sum_{i \in \mathcal{N}} \left(V_i(\boldsymbol{x}_i, \boldsymbol{y}_i) - V_i^s \right)$$

s.t. (2), (3), (4)

$$x_{ij}^{(n)} \geq 0, \; y_i^{(n)} \geq 0, \; \forall\, i, n \in \mathcal{N}, (i,j) \in \mathcal{E} \quad (8)$$

This is an optimization problem with a strictly concave objective, and a compact and convex constraint set. Hence, it admits a unique solution $(\boldsymbol{x}_i^*, \boldsymbol{y}_i^*)$. We can now proceed stating the main property of the MUPN coalitional game.

Theorem 1: The MUPN coalitional game $\langle v, \mathcal{N} \rangle$ has a non-empty core.

Proof: The proof of this property is based on the necessary and sufficient conditions derived by Bondareva and Shapley (see [11]), and follows the rationale of [20], [21]. In particular [21] proved that the multicommodity flow problem for non-linear objective functions admits a solution that lies in the core. Our proof leverages this result. ∎

Based on this result, we can devise a cooperation policy by solving the MOP problem, which yields the matrices $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$. Moreover, we can implement the payment (coin) transfers z_i^*, $i \in \mathcal{N}$ according to the shadow prices that result from the solution of this problem. This policy ensures that each subset of users (including singletons) will be satisfied enough so as not to deviate from the grand coalition. Finally, notice that the exact form of $T_i(\cdot)$ and $U_i(\cdot)$ can be determined by the implementation, as long as they have the properties discussed above. Examples are given in the sequel.

III. Architectural Blueprint

In this section, we begin with a simple example so as to illustrate the system operation and highlight the main technical challenges. Accordingly, we describe in detail each service component and explain how we tackle these raised issues. At the same time we provide pointers on how the architecture is related to the model abstraction presented in Sec. II

1) An Example and the Challenges: Assume that *Bob, Alice and John* are in proximity and wish to use their smartphones to connect to the Internet. Each device may have heterogeneous network interfaces (cellular, WiFi, Bluetooth etc). The users decide to put their devices in CoNeS mode and launch the ICSD service. The latter, initiates the discovery mechanism so as to identify the one-hop neighbors of every user who is reachable over each different network interface. Moreover, the mechanism measures the average throughput of all the available links, including the Internet connections. This information is then compiled in a star graph having an edge for each available physical link with each neighbor.

This graph is forwarded, in the form of a discovery message, by each device to the CDE service, using the closest Internet connection (either local or through other nodes). The CDE receives these messages and determines the servicing policy, i.e., the *decision graph*, which forwards to each user. This graph actually comprises the servicing policy dictated by the matrices $\boldsymbol{x}$ and $\boldsymbol{y}$, defined in Sec. II. The devices acknowledge the graph reception and apply the decision graph. Accordingly, ICSD on each device configures the local SMart Datapath (SMD) support to activate network layer forwarding. Subsequently, MUPN is up and running.

The CDE decisions are updated in a periodic fashion based on the received feedback, in the form of heartbeat messages, that are sent by the ICSD daemons. The heartbeat message format is again a star graph similar to the discovery message which, in addition, includes information about the consumed data in the previous round, the current battery levels and the data demands, i.e., whether a user wants to upload, download, communicate directly with another user, or remain idle. In particular, these messages include information about the energy consumption per transferred byte over each different link, and estimations for the respective link throughputs, i.e., parameters

$e_{ij}^s, e_{ji}^r, e_i^s, e_i^r$, and C_{ij}, C_{ui} and C_{di} that were defined in Sec. II. As these parameters change over time, the CDE is informed by the heartbeat messages and subsequently updates the decision graph when needed, which results in MUPN network re-configuration. Clearly, the heartbeat message period defines the responsiveness of CoNeS to frequent changes.

Notice that the CDE policy uses the utility and cost functions $V_i(\cdot)$ and $T_i(\cdot)$, $\forall\, i \in \mathcal{N}$, which are implementation-dependent. For example the system designer may employ logarithmic utility functions so as to achieve a proportional service allocation. Finally, this cloud-controlled system can easily retrieve the Internet usage cost for each node (parameter ρ_i, $i \in \mathcal{N}$), either directly from the nodes (ICSD), or by contacting the respective network provider (assuming there is established collaboration). This example reveals the salient *features*, beyond those described until now, that CoNeS must have. Specifically:

($F1$): *Provide a flat neighbourhood network abstraction*: Mobile devices need to support a single flat network abstraction, regardless of the possible heterogeneous intrefaces and subsequent IP configurations.

($F2$): *Fast network re-configurations*: The system should be able to update the servicing policy very fast, so as to adapt to the changing needs of the users, the varying throughput of their Internet connections, the consumption of their battery energies, etc.

($F3$): *Consistent network updates*: The previous network configuration has to be used by the neighborhood devices to receive an updated configuration. In this context the devices should quickly sync in order to deploy the new configuration only after every participant has received it.

($F4$): *Seamless transition of active flows*: Active TCP/UDP flows should not break when a new decision graph dictates changes in the gateways of the MUPN, i.e., an active flow of Bob that was reaching Internet through Alice, now has to flow through John.

To address ($F1$)-($F4$), we employed SDN to implement a programmable packet forwarding datapath on each mobile device, and used a VPN-based Internet Access Server (IAS).

2) Mobile Node System Architecture: In Figure 2 we depict the node architecture that comprises the ICSD and SMD. The former implements the required control plane logic and the latter executes datapath operations. ICSD is an mBaaS-based mobile application that interfaces with the CDE cloud backend service via a custom RESTful API. The ICSD comprises the discovery service (dcs) and the configuration service (cfs).

ICSD/dcs. This service identifies, for each node, the immediate neighbors on all available network interfaces. Each device has to start this discovery daemon that listens for neighbor probes on a globally agreed multicast IP address. These probes contain the OSI layer-2 addressing information for all the local wireless interfaces of the sender as well as a device-unique identifier. The described discovery steps are IP-specific and wireless technology agnostic, so they are common for all technologies. For each physical interface, dcs executes a throughput test for all possible links with one-hop neighbors, in order to assess their average application-level throughput. This process leverages the TLQAP protocol [38], which has

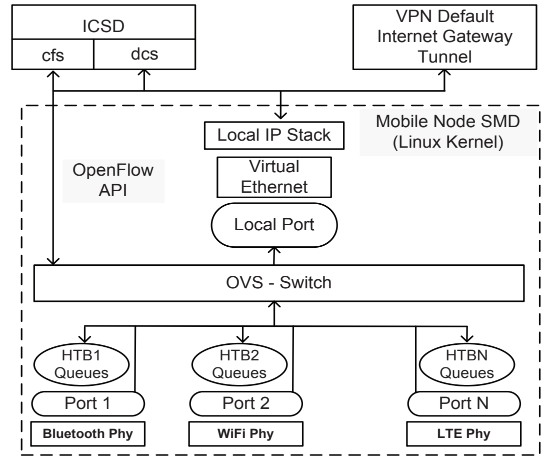


Fig. 2. Mobile node system architecture. SMD implements a forwarding element that: (i) receives configuration to relay traffic between nodes and/or the Internet, (ii) implements rate-control for egress traffic. ICSD/cfs retrieves flow statistics, determines Internet usage and sends heartbeats to CDE. SMD implements packet forwarding and supports two types of network interface roles: the (i) MUPN LAN, and the (ii) MUPN WAN.

been modified to use IP multicast and to produce the star graph based discovery messages with all the required data. This protocol has been originally developed for wireless testbeds where all the nodes are a priori known so modifications were made to integrate the described discovery step.

ICSD/cfs. This service implements a heartbeat mechanism which periodically sends the latest local star graph to the CDE with resource usage information. In this case, the links of the graph get updated with flow rule usage statistics. The latter are employed by the CDE so as to infer user activity (uploading, downloading, or being idle). When the node receives a new decision graph, the cfs identifies and applies the new role(s) of the device by appropriately updating the forwarding logic.

To dynamically shift between the different roles that each node may undertake we employ SDN to control data forwarding. Namely, if a node is assigned the role of an MUPN gateway to the Internet, the cfs implements an SDN-based Network Address Translation (NAT) scheme. More specifically, the SMD of the device is configured with packet manipulation and forwarding rules that implement NAT logic behind all local interfaces that are directly connected to the Internet. The state of the active translated flows is maintained by the cfs. For the interfaces that are used to relay neighbor traffic, appropriate flow rules are installed which rewrite the previous source and destination Ethernet addresses.

In addition to forwarding control (routing), cfs features QoS support for each different commodity (flow control), leveraging the Linux Hierarchical Token Bucket (HTB) queuing discipline [35]. HTB installs outbound queues at all network interfaces, which can be served at software-controlled rate according to the CDE decision graph. More specifically, the decision graph edges get annotated with outbound service rates towards all graph vertices (i.e., for each commodity $n \in \mathcal{N}$). Upon graph reception and analysis, each node installs a separate queue for each neighbourhood destination as well as for the Internet. Subsequently, the flow rules are generated to use the proper queue for each destination.

Datapath Support. The Smart Datapath (SMD) implements the required support on the mobile node, using the OpenVSwitch (OVS) OpenFlow datapath implementation and the HTB. ICSD configures the OVS at boot time to get control of all available interfaces. By default with OVS, the local network stack of the node is also considered as yet another network interface. The latter receives a neighbourhood-unique IP which is decided by the CDE, configured by the ICSD/cfs, and is used for D2D communication and Internet access. Figure 2 summarizes the above basic operations.

mBaaS IAS. The described support so far addresses all technical challenges but one, i.e., the seamless transition of active flows between different Internet gateways. The ICSD/cfs design enables instant change of the local SMD flow rules which allows for frequent installation of new decision graphs. On the downside, when a new decision graph assigns the Internet gateway role to another MUPN node, any active IP packet flow coming from the Internet will break immediately after the transition, because the destination IP for the MUPN ingress Internet traffic changes.

Currently, there is no support available to mitigate this problem transparently to the transport layer. One approach would be to ignore this problem and rely on the application-level protocols to deal with the connection loss. In order to improve the performance and resilience of CoNeS, we adopted a Virtual Private Network(VPN)-based Internet Access Server (IAS) approach. Note that this is an optional mBaaS service which runs on the cloud and improves performance but is not mandatory for CoNeS operation.

In this scheme, each device needs to establish a single VPN TCP tunnel with a designated VPN cloud-based IAS server to access the Internet. As a result, MUPN local gateways only have to relay VPN traffic from all devices to IAS via their local Internet connection. Each time the MUPN gateway role is assigned to another device, the respective connection to the VPN server needs to be re-established for each MUPN node, which takes place relatively fast in practice. On the other hand, users will only experience this small transition delay but active TCP connections and UDP streams will not break because the same IAS is still handling all application-level Internet traffic and there are no routing changes within the VPN overlay network. VPN-based IAS service is an important feature of the CoNeS, and a very realistic solution that can be seamlessly used in a commodity network. Choosing the right IAS server on the cloud for a MUPN deployment is important so that Internet traffic does not follow inefficient routes towards a destination. This problem is due to the use of the VPN overlay and needs to be addressed but is beyond the context of this work.

Central Decision Engine. CDE runs as a service on the cloud and is the core component of the mBaaS platform which exchanges described graphs with the MUPN nodes. The services that the CDE offers are: the (i) decision graph derivation service (dgs), the (ii) network configuration transition consistency service (ntcs), and the (iii) heartbeat service (hbs). The dgs derives the equilibrium solution of the TU coalitional game (in polynomial time) as it was described in detail in Sec. II and yields the decision graph and the coin transfers among the nodes.

These policies are based on the network state information that is sent periodically by the ICSD/cfs of each device, and collected by the CDE/hbs. This time interval is tunable in our system but, as we demonstrate in the next section, a period of 3 seconds achieves the best performance - overheads tradeoff. Based on the reported usage, CDE determines if a user needs Internet access or not. When the CDE devises a decision graph that is different than the currently active graph, it communicates it to the nodes so as to be processed by the ICSD/cfs. However, the transition to a new decision graph requires a careful synchronization of the nodes so as to ensure consistent network operation.

This is achieved by a two-stage synchronization mechanism that is implemented by the CDE/ntcs service. Namely, after the decision graph reception, the ICSD/cfs of each device reports the reception and submits a request for approval to apply it. These requests are handled by the CDE/ntcs which examines the current MUPN configuration so as to determine the order in which the different devices should migrate to the new decision graph. The main challenge during migration to a new decision graph, is that all devices need to receive the new configuration before each one starts to apply changes to its SMD. This is because SMD configurations that belong to different decision graphs might not be compatible and, therefore, MUPN network can be destroyed before all devices get the updated decision graph.

IV. Performance Evaluation

The game-theoretic CoNeS servicing policy ensures that all participants will enjoy improved performance. In practice however, sharing resources is not for free and involves additional overheads compared to independent operation. In this section, we use a specific platform configuration, based on a commodity device setup, to quantify the overheads related to (i) the heartbeat messages (CDE/ntcs), (ii) the relaying task between MUPN members, (iii) the relaying task between MUPN and the Internet, and (iv) the network re-configuration upon arrival of a new decision graph (ICSD/cfs). Notice that ConeS-specific networking operations stop at OSI layer 2 and there are no dependencies on specific network media technologies. Therefore, the performance parameters of the CoNeS tasks vary depending on the network interface and device types used in a given MUPN deployment. We also compare the proposed SDN-based forwarding with the widely used WiFi-direct approach [39].

Finally, due to lack of space, we have included only Fig. 3 that depicts numerically the theoretical performance benefits of CoNeS. It shows that the more diverse the users are, e.g., in terms of Internet access capacities or data usage costs, the more beneficial is the service for them. This diversity may refer to any device aspect such as their battery energy, Internet access capacity or cost, etc.

Experimental Setup. In order to expose the overheads and assess the limits of CoNeS-specific operations we have devised a thoroughly controlled, high performance experimental setup

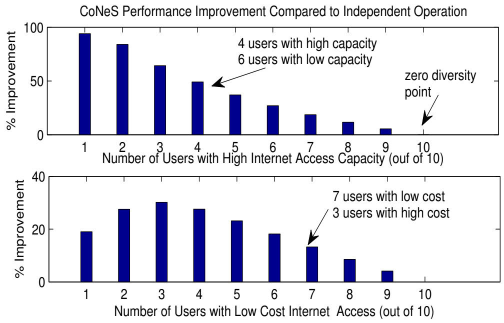


Fig. 3. Aggregate payoff improvement for a MUPN with 10 users, by solving problem MOP. Utility functions U_i are logarithmic on consumed data (r_i), energy cost functions T_i are exponential on consumed energy (e_i). Upper figure: users are identical, except the Internet access capacity. Lower figure: users are identical except their Internet access costs.

which is not affected by external parameters (unforeseen Internet delays, varying wireless throughput, poor connectivity, etc). The nodes that we used for all the experiments are single board computers (sbc) with a combination of resources that matches the performance of a modern tablet. The devices have OVS installed and run the ICSD and SMD. Internet access is available via a cable Ethernet connection of 100 Mbps, and the CDE runs on a different (physical) server located within the local Ethernet network[2]. All nodes feature a wireless 802.11n interface configured in IBSS (ad-hoc) mode [40] with fixed rate of 100 Mbps and is used to implement MUPN configurations. For the power measurements we have used the tool described in [34].

1) Node Heartbeat Overheads: First, we quantify the energy and bandwidth consumption overheads of the heartbeat mechanism, and the delay it may introduce to the system. This is also necessary so as to determine the optimal heartbeat period. Specifically, in terms of bandwidth consumption and delay, the heartbeat operation is lightweight because: (i) it requires a second(s) time period, (ii) graph representations are a few bytes, and (iii) overall operation is executed in parallel to the data forwarding operations, so delays cannot be introduced. Power consumption on the other hand is non-negligible, and depends largely on whether heartbeats induce a new decision graph. The worst case scenario is when, in every round of heartbeat messages, the CDE responds with a new network configuration.

Table I presents the power consumption of the heartbeat operation on a single node for different periods of 1 up to 5 seconds (columns), and the cases that a new configuration needs to be employed or not (rows) . The first entry is when the node is not sending any heartbeats and can be used as a reference. For each case, power consumption offsets from the reference value are presented. If the node needs to apply a new configuration for every heartbeat it sends, the CPU load and hence the power consumption increases. A period of 1 second turns out to be too frequent for a network

[2]This setup ensures that Internet links are not the bottleneck (our findings are independent of that). Also, as we demonstrate next, the location of the CDE does not affect the CoNeS performance, as such mBaaS platforms typically guarantee a near real-time response, introducing a round-trip delay of few hundreds of msecs.

TABLE I
HEARTBEAT POWER CONSUMPTION OVERHEAD T

T (sec)	idle	1	2	3	4	5
Av P. (W) (config)	$7,2$	$+0,365$	$+0,24$	$+0,21$	$+0,2$	$+0,19$
Av P. (W) (no config)	$7,2$	$+0,25$	$+0,17$	$+0,16$	$+0,14$	$+0,11$

TABLE II
ENERGY CONSUMPTION IN RELAYING OPERATION

Operat. modes	Idle(node)	Idle (NIC)	Relay(node)	Relay(NIC)
Av P. (W)	$7,2$	$0,58$	$+1,3$	$+0,9$
Cons. (J/ Mb)	–	–	$0,12$	$0,08$

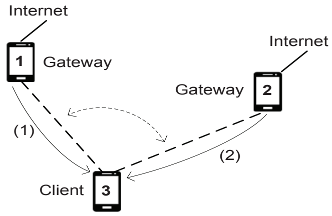
reconfiguration to complete, and it also introduces a significant power consumption overhead. On the other hand, a 5-second period is at the upper bound of responsiveness without actually providing any serious power consumption benefits.

Findings:*CDE/hbs does not introduce delay in the system, does not consume significant bandwidth, and a period of 3 seconds has been found to be optimal in practice, introducing an additional 2.22% - 2.95% energy consumption per device. Typically cloud backend service response in mBaaS architecture is in the order of a few msecs, so CDE cannot induce delays that would affect the CoNeS operation. We use 3 secs as the heartbeat period in the experiments that follow.*

2) MUPN Relaying Overheads: The task of relaying, besides the bandwidth, consumes battery energy, and computing resources for heartbeats and data forwarding. One factor that has the potential to overload a CPU (and even affect throughput) is the number of active flow rules concurrently installed in OVS, as it increases the per-packet overhead of lookup/match operations. Generally, it is hard to predict how OVS performance will scale with the number of these rules, as they are frequently evicted and reinstalled based on the traffic pattern. However, since CoNeS targets the (mobile) wireless domain, the current upper bound that is reasonable to support is $100Mbit$ or $\sim 200K$ 64-byte packets per second (Kpps) of switching capacity. For the modern CPU capabilities, a $200Kpps$ OVS forwarding overhead has been found to be minimal [37].

On the other hand, in addition to network interfaces, CPU does spend energy to support forwarding in CoNeS. In order to measure this overhead, we configured a decision graph for a tandem network of three nodes ($1 \rightarrow 2 \rightarrow 3$), where node 1 sends data to node 3, via 2 (relay). We have configured the transmission power of each node to a fixed value of 10dBm, and used an isolated frequency channel for all the transmissions. The relaying device uses the same 802.11 interface for both Rx and Tx; hence the direction of the main traffic volume is not relevant for the power consumption (same TX for any direction). This the simplest MUPN deployment which requires the relaying task role. Notice that all CoNeS services are active during the measurements that follow.

First, in order to get a reference, we measured the power consumption of an sbc node and its NIC in idle operation and found it on average to be 7.2 and 0.58 Watts, respectively. Subsequently, we measured the energy consumption of the relaying operation of node 2. Wireless media power consumption depends on the throughput [36] which in turn



T (sec)	–	30	25	20	15	10	6
Av. Mbps	107	$92,5$	$88,7$	87	$81,7$	$68,5$	$51,8$
Delay(Sec)	157	186	193	197	210	245	324
Av. P. (up) (W)	$+0,58$	$+0,55$	$+0,55$	$+0,49$	$+0,48$	$+0,42$	$+0,41$
Av. P. (dn) (W)	$+1,38$	$+1,35$	$+1,35$	$+1,29$	$+1,28$	$+1,22$	$+1,21$
Cons. (up) (J/MB)	$0,045$	$0,051$	$0,054$	$0,048$	$0,051$	$0,052$	$0,066$
Cons. (dn) (J/MB)	$0,1$	$0,125$	$0,13$	$0,127$	$0,134$	$0,15$	$0,19$

Fig. 4. Gateway switching overhead: user 3 is a client being relayed either by user 1 or user 2. Rows show aggregated service rate, delay and average upload/download energy consumption for different gateway switching intervals T, for downloading a 2GB file. Fixed additional overheads are: i) Average graph install delay (after reception) is measured to 1 msec. ii) TCP VPN recover after exchange (average): 1 second to detect exchange, 2.3 sec to re-initiate connection, 3.3 seconds total stall. The minimum threshold that allows a full cycle of graph downloads to complete was found to be 6 secs.

is affected by the wireless channel conditions, the distance of the nodes, etc. For these experiments, a wireless throughput of $100Mbps$ was achieved on average. In Table II the sbc and NIC power consumption during the aforementioned relaying operation is presented as a reference to idle. Moreover, average consumption rate of energy per relayed MByte of data is reported. The traffic is generated with the iperf tool and saturates the wireless capacity ($48Mbit$).

Findings: *The SDN operation of CoNeS does not induce significant CPU performance overheads (below 2% occupancy [37]). Regarding power consumption it is observed that 802.11 Tx operations account for* 65% *of the required power budget and the rest is consumed by the CPU and memory copies.*

3) Network Reconfiguration and Internet Relay Overheads: In the experiments that follow we measure the throughput and power consumption overheads for different network reconfiguration frequencies where different gateways relay traffic to the IAS. Accordingly, the previous 3-node CoNeS setup is changed and now features two MUPN gateway nodes (1 and 3) that periodically serve the client (2), Fig. 4. Table 4 presents forwarding performance, average delay and the aggregate energy consumption of the two gateways separately for uploading and downloading (via TCP) a 2 GByte file between node 2 and a file server accessed over IAS in the local network. The first row is the frequency of network reconfiguration intervals in seconds. The second row is the average Mbps service rate of file transfer, and the third depicts the average completion time of the transfer. The next two rows provide the aggregated average power consumption (offsets from idle) of the gateways for the upload and download of the file and the last two rows have the corresponding energy consumption per MByte of data. The reference is when a single gateway serves the entire file transfer (first column)[3].

In this experimental setup the relaying of Internet traffic involves seperate interface technologies which reflects the typical configuration of MUPN gateways in CoNeS. Often 3G or LTE is expected to be used instead of Ethernet but even these technologies have different power consumption demands from 802.11 for data Rx/Tx. Since heterogeneous Tx operations have different power consumption, uploads and downloads from the Internet are shown to have a different impact, while the rest of the performance parameters are not affected by the traffic direction. Average energy consumption rate per MByte transfer is not seriously affected by the frequency of gateway changes. This is because the MUPN, while it does need more time to complete the transfer with more frequent network reconfigurations, it requires less average power and hence results in similar overall energy consumption.

On the other hand there is an impact to user experience since service rate can significantly vary with frequent updates, but even in the worst case it remains at an acceptable level for the mobile user. In our experiments, the interval of 20 secs provided the best balance between the discussed trade-offs. Finally we have verified, that the client active TCP/UDP flows are oblivious to gateway switching when they run on top of a VPN. Hence, the observed delays are related to the re-establishment of the VPN tunnel and in the case of TCP, the congestion control response to a sudden throughput drop when a reconfiguration is in progress.

Findings: *MUPN Network reconfiguration has acceptable performance delays which are induced only when the MUPN gateway is changed and as long as the lowest period of a gateway switch is no less than* $20secs$. *If gateway switch frequency is not violated, the aggregated energy consumption of cooperating gateways is similar to the energy required by one gateway to perform the same data transfer.*

4) CoNeS Vs WiFi-direct based MUPN: WiFi-direct implements easy-to-configure MUPNs and it is the main approach used in current commercial solutions [6]. In [40] the internals of WiFi-direct are described and a detailed comparison regarding flexibility versus WiFi IBSS is presented. Since CoNeS MUPN forwarding is cloud-controlled, it requires basic datapath support on mobile devices and does not need to rely on the WiFi direct sophisticated operations. In order to compare the performance of these alternatives, we have deployed WiFi-direct on the sbc nodes and repeated the previous experiment ($IV-2$). Regarding forwarding performance WiFi direct relay achieved an average of $16,7Mbits$ versus $48Mbit$ of WiFi-IBSS which is expected due to the opportunistic power saving and notice-of-absence (NoA) functions of the former[39]. Moreover, it is observed that WiFi direct virtual access point has a fixed power overhead so the average idle power is $7,6W$ versus $7,2W$ required by CoNeS. Finally, WiFi direct needs hardware support and the respective WiFi interface requires 50% more energy to operate at full speed.

V. Related Work

1) UPNs: A key issue in mobile UPNs is the efficient allocation of user resources. Reference [15] proposed an energy-

[3]Notice that these numbers do not include the idle energy consumption of each gateway device, assuming that the devices are switched on independently of their participation in CoNeS.

prudent architecture for mobile hotspots, and [16] presented a request admission scheme for maximizing hosts' revenue. Client and host roles for each user, should be determined based on battery energy [17]. These works assume that users have strong social ties [19] and hence collaborate, or they do not consider this issue. This was studied in [32], but only for one-hop architectures, and not using a game-theoretic analysis. Also, [31] designed an incentive mechanism which however did not consider the possibility of subgroups creation. On the contrary, CoNeS policy is derived by the solution of a coalitional game (subgroups are allowed to emerge), and more importantly comes with a prototype implementation and detailed evaluation.

2) SDN and Mobile Networks: SDN has been mainly employed for network virtualization, data center and cloud computing architectures, e.g., see [22], [24]. Its application to IEEE 802.11/16 networks was proposed in [23], and [25]. In this work, we go a further step considering SDN-enhanced mobile devices [27] so as to enable the collaborative consumption of mobile Internet resources. Interestingly, such bottom-up architectures bridge heterogeneous (e.g., WiFi and cellular) and independent (from different providers) networks, and open novel research and business opportunities. SDN-enabled wireless mesh networks were studied before, focusing on efficient mobility management [26]. These works consider fixed nodes, and do not study the collaboration issues that arise when the nodes are owned by different entities.

There are only few works discussing networking with SDN-enabled mobile devices [18]. Among them, [28] presents an SDN implementation over wireless ad hoc networks at the application layer, and [29] studies SDN-based tethering solutions for one-hop architectures, focusing on authentication mechanisms. On the other hand, a simulation-based study [30] proposes a cloud-based SDN controlled architecture for mobile ad hoc networks. Here, we design and implement a cooperative Internet access architecture that takes into account monetary and energy consumption costs for each participant.

VI. Conclusions

The proposed architecture is motivated by the concept of collaborative consumption [33], which, we believe is very appropriate for wireless communications and user-provided networks in particular. Interestingly, several innovative startups [6], [5], [7], have already launched such services. At the same time, such models are embraced by traditional network operators which on the one hand aspire to offload their congested networks and, on the other hand, identify novel business opportunities in this area [12], [4]. CoNeS is a prototype implementation of an innovative cloud-control, SDN-enabled UPN system with emphasis on mobile devices. It leverages an efficient and group-rational collaboration policy that incentivizes the users to participate and exchange with each other *bits and coins*, creating a win-win situation through a network effect. We believe that our game-theoretic analysis, detailed system design and targetted performance evaluation study, lays the foundations for the development of a new class of such UPN systems, and opens exciting new research directions.

VII. Acknowledgment

This work has been supported by the European Commission through the FP7 "CODELANCE" IAAP, No. 285969.

References

[1] Cisco, "Global Mobile Data Traffic Forecast, 2012-2017", 2013.
[2] ITU Report, "Measuring the Information Society", Oct. 2013.
[3] R. Sofia, and P. Mendes, "User-Provided Networks: Consumer as Provider", *IEEE Comm. Mag.*, vol. 46, no. 12, 2008.
[4] FON: http://www.fon.com
[5] Karma: http://www.yourkarma.com/
[6] Open Garden: http://www.opengarden.com/
[7] M-87: http://www.m-87.com/
[8] CTIA [Online] "Consumers Use Average 1.2 GB a Month", Nov. 2013.
[9] NY Times, " iPhones Overload AT&T Network", Sept., 2009.
[10] M. Afanasyev, et al., "Usage Patterns in an Urban WiFi Network", *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, 2010.
[11] R. Myerson, "Game Theory: Analysis of Conflict", *Harvard Press*, 1997.
[12] BeWiFi [Online]: http://www.bewifi.es/
[13] M. Kodialam, et al., "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks", *ACM MobiCom*, 2005.
[14] J. Huang, et al., "A Close Examination of Performance and Power Characteristics of 4G LTE Networks", *In Proc. of ACM MobiSys*, 2012.
[15] A. Sharma, et al., "Cool-Tether: Energy Efficient On-the-fly WiFi Hotspots using Mobile Phones", *ACM CoNext*, 2009.
[16] N. Do, et al., "CrowdMAC: A Crowdsourcing System for Mobile Access", *ACM Middleware*, 2012.
[17] E. Jung, et al., "User-Profile-Driven Collaborative Bandwidth Sharing on Mobile Phones", *ACM MCS*, 2010.
[18] J. Lee, et al., "meSDN: Mobile Extension of SDN", *ACM MCS*, 2014.
[19] L. Keller, et al., "MircoCast: Cooperative Video Streaming on Smartphones", *ACM Mobile CCR*, vol. 16, no. 4, 2012.
[20] C. Singh, et al., "Cooperative Profit Sharing in Coalition-Based Resource Allocation in Wireless Networks", *IEEE/ACM ToN*, vol. 20, 2012.
[21] E. Markakis, and A. Saberi, "On the Core of the Multicommodity Flow Game", *Decision Support Syst.*, vol. 39, 2005.
[22] M. F. Bari, et al., "Data Center Network Virtualization: A Survey", *IEEE Comm. Sur. and Tut.*, vol. 15, no. 2, 2013.
[23] K.-K. Yap et al., "Blueprint for Introducing Innovation into Wireless Mobile Networks", *in Proc. of ACM VISA*, 2010.
[24] K. Pentikousis, et al., "MobileFlow: Towards Software-Defined Mobile Networks", *IEEE Comm. Mag.*, vol. 51, no. 7, 2013.
[25] J. Vestin, et al., "CloudMAC: Towards Software Defined WLANs", *ACM Mobile CCR*, vol. 16, no. 4, 2012.
[26] A. Amokrane, et. al., "Online Flow-based Energy Efficient Management in Wireless Mesh Networks", *IEEE Globecom*, 2013.
[27] K.-K. Yap, et al., "Making Use of All the Networks Around Us: A Case Study in Android", *ACM CellNet*, 2012.
[28] P. Baskett, et. al, "SDNAN: Software-defined Networking in ad hoc Networks of Smartphones", *IEEE CCNC*, 2013.
[29] M. Santos, et al., " Software-defined Networking Based Capacity Sharing in Hybrid Networks", *IEEE ICNP*, 2013.
[30] I. Ku, et al., "Software-Defined Mobile Cloud: Architecture, Services and Use Cases", *in Proc. of IWCMC*, 2014.
[31] G. Iosifidis, et al., "Enabling Crowd-sourced Mobile Internet Access", *in Proc. of IEEE Infocom*, 2014.
[32] T. Yu, et al., "INDAPSON: An Incentive Data Plan Sharing System Based on Self-Organizing Network", *in Proc. of IEEE Infocom*, 2014.
[33] R. Botsman, and R. Rogers, "What's Mine Is Yours: The Rise of Collaborative Consumption", *HarperBusiness*, 2010.
[34] S. Keranidis, et al., "Online Energy Consumption Monitoring of Wireless Testbed Infrastructure", *ACM WinTech*, 2013.
[35] Linux HTB [Online] http://luxik.cdi.cz/ devik/qos/htb/
[36] J. Huang, et al., "A Close Examination of Performance and Power Characteristics of 4G LTE Networks", *ACM MobiSys*, 2012.
[37] L. Rizzo, et al., "Transparent Acceleration of Software Packet Forwarding Using netmap", *IEEE Infocom*, 2012.
[38] D. Syrivelis, et al., "TLQAP: A Topology and Link Quality Assessment Protocol for Efficient Node Allocation on Wireless Testbeds", *ACM WinTech*, 2009.
[39] Daniel Camps-Mur, et al., "D2D communications with WiFi Direct: overview and experimentation", *IEEE Wireless Communications*, vol. 20, no. 3, 2013.
[40] WiFi-direct Vs IBSS http://www.thinktube.com/tech/android/wifi-direct.