# Upgrading Wireless Home Routers for Enabling Large-Scale Deployment of Cloudlets

Christian Meurisch[1(✉)], Alexander Seeliger[1], Benedikt Schmidt[1], Immanuel Schweizer[1], Fabian Kaup[2], and Max Mühlhäuser[1]

[1] Telecooperation Lab, Technische Universität Darmstadt, Darmstadt, Germany
{meurisch,seeliger,schmidt,schweizer,max}@tk.tu-darmstadt.de
[2] Peer-to-Peer Systems Engineering Lab, Technische Universität Darmstadt, Darmstadt, Germany
fkaup@ps.tu-darmstadt.de

**Abstract.** Smartphones become more and more popular over recent years due to their small form factors. However, such mobile systems are resource-constrained in view of computational power, storage and battery life. Offloading resource-intensive tasks (aka *mobile cloud computing*) to distant (e.g., *cloud computing*) or closely located data centers (e.g., *cloudlet*) overcomes these issues. Especially, cloudlets provide computational power with low latency for responsive applications due to their proximity to mobile users. However, a large-scale deployment of range-restricted cloudlets is still an open challenge. In this paper, we propose a novel concept for a large-scale deployment of cloudlets by upgrading wireless home routers. Beside router's native purpose of routing data packets through the network, it can now offer computing resources with low latency and high bandwidth without additional hardware. Proving our concept, we conducted comprehensive benchmark tests against existing concepts. As result, the feasibility of this concept is shown and provide a promising way to large-scale deploy cloudlets in existing infrastructures.

**Keywords:** Wireless home router · Mobile cloud computing · Cloudlet · Smartphones · Offloading · Edge computing

## 1 Introduction

Many mobile services require complex computations, e.g., voice processing for a dialogue system or image processing for an augmenting application. Such services need to address the performance requirements while considering the short battery life of mobile devices [22]. To address this challenge most service providers rely on *mobile cloud computing* [10,12]: resource-intensive tasks are offloaded to distant servers [18]. However, latency and network traffic are one of the downsides of this approach. Therefore, mobile cloud computing is best suited for applications with need of high availability and global view like social networks. Applications with high computing requirements and the need for responsiveness are not perfectly suited for the cloud computing approach.


S. Sigg et al. (Eds.): MobiCASE 2015, LNICST 162, pp. 12–29, 2015.
DOI: 10.1007/978-3-319-29003-4_2

An alternative to mobile cloud computing is the use of *cloudlets* [24]. Cloudlets are small-scale servers which are distributed over the environment. Mobile devices connect to nearby cloudlets to distribute computation tasks and benefit from a one-hop latency over wireless communication technologies [24]. Thus, cloudlets offer a promising tradeoff between performance gain, low network traffic and especially low latency. This makes cloudlets especially relevant for applications with high computation and responsiveness requirements like face or object recognition with the fast processing of big sensor data [17].

A combination of cloud computing and an extensive dissemination of cloudlets would address the requirements of various types of mobile services. While mobile cloud computing is well-established, there is no extensive dissemination of cloudlets, yet. Two different approaches for the realization of cloudlets have been proposed. First, a grassroots perspective, focusing on the deployment by local businesses (e.g., cafes or shopping malls) which step-by-step evolves to a large-scale infrastructure offered and maintained by the businesses [24]. The second perspective is the integration of cloudlets into Internet's routing infrastructure at the gateways of ISPs [6] or by combining cloudlets and wireless mesh networks, deployed in hotspots [16]. However, each concept requires the deployment of additional computing hardware by different entities resulting in deployment and operation costs. Therefore, the realization of a dense and economic cloudlet infrastructure is still an open challenge.

In this paper, we propose a router-based cloudlet concept to realize an extensive dissemination of cloudlets based on existing infrastructure. Our concept promises a dense distribution of cloudlets in many countries while avoiding unpredictable economic risks for the involved parties. On average, 73.0 % of EU households [27] and 75.6 % of US households [13] have access to the Internet in 2011, many of them using wireless routers to connect to the Internet. Hence, in our view, wireless home routers are well-suited for a large-scale, dense and economic infrastructure [19] to offload computational tasks from mobile systems.

Beside router's native purpose of routing data packets through the network, our concept offers its computing resources to mobile devices without deploying additional hardware. In other words, we treat a wireless home router as cloudlet with both networking and computing capabilities. Mobile devices connecting to such nearby located router via wireless technologies (i.e., WLAN) benefit from offloading capabilities with low latency and high bandwidth. We imagine two use cases; *on the one hand*, responsive applications (e.g., face recognition), that require low latency and fast responses, can be directly served by the router maintaining a soft state (i.e., temporary cache). *On the other hand*, contextual applications, that require historical data in some circumstances, can leverage the router as intermediate layer to the cloud which preprocesses data and, thus, reduces network traffic for connections with high latency.

The concept needs to be assessed with respect to two main aspects: (1) feasibility – what is necessary to use current state of the art routers as cloudlet (2) performance – do the limited computational capabilities of routers justify the effort compared to other techniques. In this paper, we report the concept, assess the feasibility and conduct performance benchmark tests covering energy consumption, resource usage, network traffic, latency and processing time for the following approaches: (1) local mobile processing, (2) cloudlet processing, and (3) cloud computing.

In summary, the contributions of this paper are twofold:

**Concept for Router-based Cloudlets.** We propose a novel concept for solving large-scale, dense and economic deployment issues of range-restricted cloudlets utilizing existing wireless home routers for mobile cloud computing. In detail, a nearby located wireless home router can now offer computing resources to mobile devices without deploying additional hardware, beside router's native purpose of routing data packets through the network.

**Performance Benchmarks.** Proving our concept we conducted comprehensive benchmark tests against existing concepts like cloud or common cloudlets. These tests cover measurements of energy consumption, resource usage, network traffic, latency, and processing time from the viewpoint of a mobile device.

The remainder of this paper is organized as follows. First, we give an overview of related work and work out open issues. Second, we report the concept of upgrading wireless home routers as cloudlet for enabling large-scale deployment on existing infrastructure. After reporting, the experimental setup and methodology is described. The paper closes with benchmark report, discussion of the benchmark results, and conclusion.

## 2 Related Work

The need for offloading computational tasks and storage from resource-constrained mobile systems (e.g., smartphones, Internet-of-things devices) introduced *mobile cloud computing* [1,12,15,21] or *cyber foraging* [3,23] about fifteen years ago. Since then, various offloading approaches regarding networked computing infrastructures (e.g., *cloud computing* [18], *cloudlets* [24,25,30], *fog computing* [4,28]) and offloading strategies (e.g., MAUI [9], CloneCloud [7]) were proposed to find a tradeoff between performance, latency and network traffic.

In the following we revisit different strategies to realize computation tasks with mobile devices [10]:

*Mobile Computing.* Mobile devices are able to process data locally without latency issues. However, due to their small form factor and high mobility mobile devices have limited resources, e.g., battery life, storage and computational power [22].

*Cloud Computing.* Resource-intensive tasks are offloaded via the internet from mobile devices to centralized resourceful data centers, the *cloud*. The cloud is a highly scalable computing and storage infrastructure hosted by cloud providers (e.g., Google, Amazon, and Salesforce) [18]. A cloud serves and stores personal data of hundreds or thousand users at a time. Security, privacy and trust are highly critical points. However, clouds are distant to

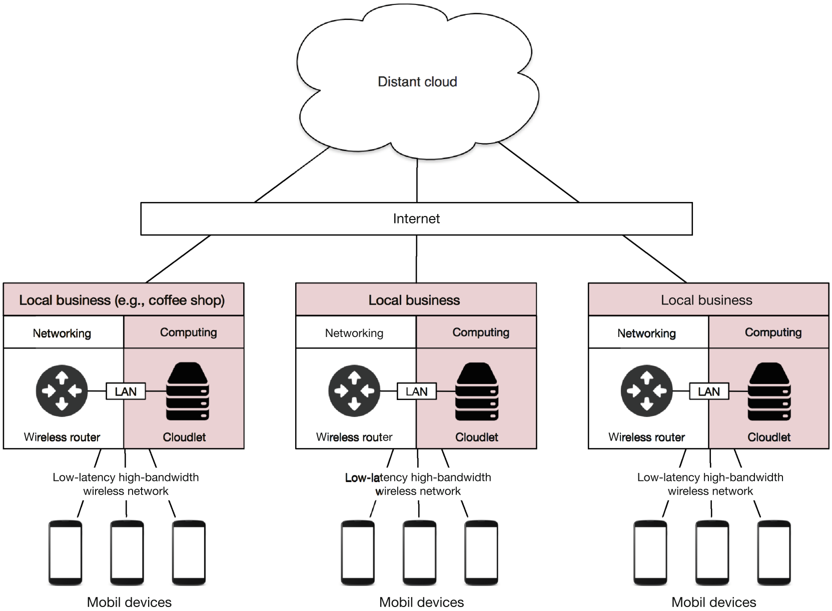



**Fig. 1.** Original *cloudlet* concept firstly proposed by [24] and deployment challenges for a comprehensive computing network infrastructure are marked in red (Color figure online).

mobile users and have too long WAN latency for responsive applications. But they are well-suited for applications requiring a global view or historical data. Moreover, only few data centers are deployed in the world with high building and operational costs.

*Cloudlet.* Resource-intensive tasks can also be offloaded from mobile devices via wireless technologies (e.g., WLAN) to a *cloudlet* (cf. Fig. 1), a proximate decentralized computing infrastructure hosted by a local business (e.g., coffee shop) [24] or ISPs [6]. It provides low latency due to its proximity to mobile users and high bandwidth. Thus, cloudlets are well-suited for real-time responsive applications like face, gesture or object recognition that only need temporary caches [20]. Cloudlets only need to serve few users at a time. However, a large-scale deployment of current approaches is difficult due to their range restrictions and their high costs.

In summary, we identified three issue groups that need to be considered in terms of mobile cloud computing: limited mobile resources (e.g., battery life, storage, computational power), communication issues (e.g., latency, bandwidth, network traffic) and remote processing issues (e.g., security, privacy, ownership, scalability, deployment and operational costs). Focussing on *cloudlets*, the first two issue groups are overcome by that approach [24], i.e., cloudlets offer offloading resource-intensive tasks with low latency and high bandwidth to overcome limited resources on mobile devices. However, we see an open challenge in the last issue group for cloudlets, especially a deployment concept for establishing a comprehensive and dense computing infrastructure with cloudlets is still missing. Figure 1 shows the original cloudlet concept firstly proposed by Satyanarayanan et al. in 2009 [24], where cloudlets are deployed in local businesses like coffee shops or shopping malls. Since then, a large-scale and economic deployment concept does not exist. We also mark the key components in red (cf. Fig. 1) that are responsible for failure of a large-scale deployment, namely, the need of deploying additional computing hardware and the deploying in local businesses which are not geographical dense and comprehensive distributed.

In this paper, we address the main issue of cloudlets: a *large-scale deployment*. A comprehensive, dense and highly available but economic cloudlet infrastructure is essential to make this approach suitable for everyday life.

## 3 Concept for Router-Based Cloudlets

We propose router-based cloudlets to offload computations from mobile device. Like most offloading techniques, we strive for saving resources and increase responsibility for a better user experience. Our concept benefits from the dense distribution of wireless routers which will result in a large-scale, dense and economic cloudlet infrastructure without the need for new infrastructure invests. This approach can complement existing cloudlet deployment concepts (e.g. , local business, ISP gateway). A router-based cloudlet infrastructure will increase the overall awareness of cloudlets and their benefits. This might also facilitate the existing deployment concepts with more computational power. In the following, we specify our concept for router-based cloudlets. *First*, we investigate the feasibility on device level: can routers be used as cloudlets? *Second*, we investigate the creation of an infrastructure based on cloudlets of routers to be used by mobile devices. *Third*, we consider the community environment of the concept and address legal and social challenges of the process.

### 3.1 Device (Router)

Inspired by active network research [14, 29], our goal is to leverage computational power from wireless home routers. While these routers are currently only used as network devices, we also want to use them as cloudlets, i.e., providing computational power in the network (cf. Fig. 2). To add this functionality, a basic software update or firmware customization is sufficient for many routers (see the evaluation section of this paper for details). This process can open a socket for computational task requests. In the future the cloudlet functionality could be integrated from manufactures or ISPs that provide routers to customers.

Mobile devices can simply connect to wireless routers and benefit either from high-bandwidth to the cloud for contextual applications that need global or historical view (Internet latency) or from computational power of routers for

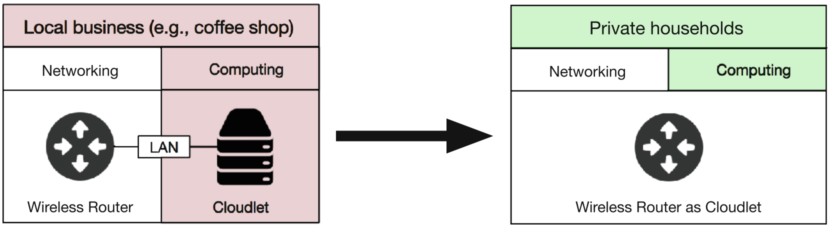



**Fig. 2.** Original *cloudlet* concept firstly proposed by [24] (*left*) and our approach for enabling a large-scale deployment of cloudlets by upgrading wireless home routers (*right*).

responsive applications (LAN latency). Depending on the need of mobile applications, requests are sent to different endpoints: benefiting from cloud, requests are addressed to cloud's IP address or hostname. These requests are automatically forwarded by routers (*sharing of high-bandwidth Internet connection*).

A specific benefit of routers for the intended purpose is that they are always online (Internet, power grid), have a low latency (near located), and a high bandwidth (WLAN). Drawbacks of routers are their low range and low computational power. It is necessary to address the low range with respective infrastructure protocol (see next paragraph). For the low computational power, a further investigation of the performance of modern routers considering the benefit of low latency is required (see evaluation section).

### 3.2 Infrastructure

The router-based cloudlets need to be accessible as an infrastructure to be used by mobile devices. As already mentioned, we assume that the router-based cloudlet infrastructure will be complemented by dedicated cloudlets. All cloudlet types need to be integrated on one infrastructure.

The most important challenge is how to access the devices and structure the use. Next to the discovery, challenges with respect to congestion handling, failure handling and handover need to be addressed. For most problems, similar challenges have already been addressed for cellular networks, therefore we plan to transfer existing solutions to the cloudlet domain. To realize discovery, we plan to build a router guest network with all routers using a similar SSID. If computations are not finalized before a device leaves the range, handover mechanisms are required. Additionally, failure handling mechanisms, which also take account of the cloud as a fallback solution, will be considered.

### 3.3 Community

The real world deployment of routers as cloudlets has to address different social and legal challenges. In the following, we consider three important aspects:

**Willingness and Activation.** *How could we motivate household owners to upgrade their private routers as cloudlet and share these resources with others?* We believe in a "give-and-take concept" similar to established concept for free mobile Internet like in the research project *Mobile ACcess*[1] by university RWTH Aachen or in the commercial sharing product "WLAN TO GO"[2] by Internet provider Deutsche Telekom. Inspired by these concepts, mobile users sharing and upgrading their own home routers as cloudlets are allowed to connect to nearby upgraded cloudlet routers of other participants and benefit from these offloading resources. Activation or upgrading router to a cloudlet could be simply software-based done by either a firmware update through owners or already customized firmware of manufactures or Internet providers.

**Security and Privacy.** *How could both the home network and its resources be secured? How could we protect the privacy of members of the households and participating mobile users?* Modern customary routers provide the possibility to setup a home network and an isolated guest network (including our test router Asus RT-AC87U[3]). We could utilize such software-based separation of two networks to isolate home network and the public accessible network providing cloudlet functionalities. In future, manufacturers can think about hardware-based separation of both networks and provide dedicated cloudlet functionalities inherently.

**Legal Issues and Digital ID.** *How is the legal position in crime situation by sharing resources? How could we identify and authorize users to allow them access to routers' resources?* The legal position in crime situation is still an open question in our concept or in common cloudlet deployment (e.g., in a coffee shop) and depends from country to country. However, we propose an authorization and an authentication mechanisms to get access to other routers. First, a household owner upgrades and shares his wireless home router to other participants. As consequence, he gets the right to access other routers of participants (*authorization*). Second, his upgraded router and his mobile devices exchange device IDs and a digital ID similar to an authentication token that is unique for each household. Connecting to other routers this token is sent for checking participating users (*authentication*). A centralized instance (e.g., Internet provider) need to maintain digital IDs and access rights. In this way, every usage of other routers are personalized and the use can be traced back to specific household and natural person for law cases.

Nevertheless, to show the feasibility and the high potentials of our concept to build a dense, comprehensive and economic computing infrastructure that is highly available, we conducted benchmark tests against current cloudlet concepts

[1] http://mobile-access.org (accessed 2015-08-10).
[2] http://www.telekom.de/privatkunden/zuhause/zubuchoptionen/internet-optionen/hotspot/wlan-to-go (accessed 2015-08-10).
[3] http://www.asus.com/Networking/RTAC87U/specifications (accessed 2015-08-10).

and clouds. The tests should show that routers suffice the performance requirements; in other words, it is first necessary to show that the comparatively weak computational power of a home router is of no consequence in processing time of offloading tasks because of benefiting from the low latency due to the proximity and high bandwidth over wireless LAN technologies. In the next section we describe the experimental setup, before we report and discuss the results of conducted benchmark tests.

## 4 Experimental Setup

In this section the experimental setup, i.e. hardware components and measurement methodology is described. Our experimental setup consists of a mobile device and different offloading systems (i.e., cloud, cloudlet, our router-based cloudlet) for comparing. Our goal is to show that modified wireless routers match performance requirements of a cloudlet. For that, we measure energy consumption and resource usage on mobile device as well as task completion time divided into network delay and processing delay when offloading computational tasks to each system.

### 4.1 Hardware

**Mobile Device.** We use a LG Nexus 5 smartphone with quad-core ARM processor (Qualcomm Snapdragon 800) which each core running at 2.26 GHz, 2GB memory and 16GB storage (cf. Table 1). The operating system is updated to the recent standard Android 5.1.1 ROM, namely Lollipop. All background services not required for running the operating system are disabled. Nexus 5 is equipped with 2300mAh Lithium polymer (LiPo) battery by default. We chose that smartphone because it includes all electronics required for measuring the battery voltage and the current flowing from battery to the device. Thanks to integrated MAX170485 fuel-gauge chip[4] that provides high-accuracy voltage measurements and battery level estimation. It has a resolution of 1.25 mV with an error of 7.5 mV. Accurate enough for our measurement purpose to detect differences between the single offloading use cases. Nexus 5 is also equipped with an IEEE 802.11 a/b/g/n/ac wireless transmitter and supports all digital cellular networks ranging from 2G (GSM) to 4G (LTE). As result, this smartphone is able to offload tasks over Internet to the cloud as well as over wireless LAN to cloudlets or wireless routers.

**Cloud.** As cloud backend, we deployed three Amazon Elastic Compute Cloud[5] (EC2) instances, namely *c3.large*, hosted at different countries with different pricing models (cf. Table 2). Each instance provides two compute units, 4GB

[4] http://www.maximintegrated.com/en/products/power/battery-management/MAX17048.html (accessed 2015-08-10).

[5] https://aws.amazon.com/en/ec2 (accessed 2015-08-10).

**Table 1.** Smartphone specifications

| | |
|---|---|
| Model | LG Nexus 5 |
| Processor | Quad-core 2.26 GHz Qualcomm Snapdragon 800 (ARM) |
| Memory | 2 GB RAM |
| Storage | 16 GB |
| OS | Android v5.1.1 (Lollipop) |
| Power | 3.8 V, 2300 mAh LiPo battery (8.74 Wh) |
| WLAN | IEEE 802.11 a/b/g/n/ac, dual-band (2.4/5 GHz) |
| Network | GSM (2G)/UMTS (3G)/HSDPA (3.5G)/LTE (4G) |

RAM and 32GB SSD storage. For deploying our processing code written in JavaScript, we utilize Amazon Elastic Beanstalk[6] to automatically setup an appropriate runtime environment (i.e., Linux, NodeJS).

One focus of our benchmarking is task completion time, not only computational power is relevant, but also network latency. Thus, we test three different located clouds (US West, central Europe, Asia Pacific) with same computational power to get the impact of network latency [8]. In Table 2, linear distances between our measurement conducting location (Darmstadt, Germany) and the cloud data centers are listed. Only considering the distance, clouds are at a disadvantage compared to nearby located cloudlets regarding latency because of physical constraints: information cannot propagate faster than the speed of light ($\sim 3 \cdot 10^8$ m/s) when dealing with long distance. While light is able to use beeline, information travels through deployed glass fiber infrastructure with a slightly longer path (let's assume 20 % longer) and with refractive index of about 1.5. Simple mathematical calculations provide us the result how long light need only to travel via air and via glass fiber to the cloud and back (cf. Table 2, RTT). As result of this simple calculations, we can say latency cannot be ignore when talking about distant clouds. Regarding costs: building and operating a data center is extremely expensive for a cloud provider; that is the reason why only few data centers exist worldwide. As user, the setup of a cloud is free but using resources are expensive, as you see the pricing model (costs per working hour) in Table 2.

**Cloudlet.** As cloudlet we use a desktop computer with quad-core x64 processor (Intel Core i7) running each core at 3.6 GHz, 16 GB RAM, 1TB HDD storage and linux-based operating system (Table 3). The same processing code as used for the cloud is also used for the cloudlet. The cloudlet is placed in the near of the mobile device as well as has one-hop latency and LAN bandwidth. Deploying such a cloudlet server would cost about 1,000$ acquisition cost and about 11 Cent operational costs per hour. Considering these sums of money and range

[6] https://aws.amazon.com/en/elasticbeanstalk (accessed 2015-08-10).

**Table 2.** Cloud specifications: Amazon EC2 instances (as of 08/2015)

| | US West (Oregon) | EU (Frankfurt) | Asia Pacific (Sydney) |
|---|---|---|---|
| Instance | c3.large | | |
| Processor | 2 vCPU (Intel Xeon E5-2680 v2 2.8 GHz) | | |
| Memory | 4 GB RAM | | |
| Storage | 2x 16 GB SSD | | |
| OS | 64 bit Amazon Linux 2015.03 v2.0.0 | | |
| Distance (beeline) [km] | 8,500 | 30 | 16,500 |
| RTT (air) [ms] | 57 | 0.2 | 110 |
| RTT (glass fiber) [ms] | 85 | 0.3 | 165 |
| Costs (asset/working) [$] | –/0.105 | –/0.129 | –/0.132 |

**Table 3.** Cloudlet specifications

| | |
|---|---|
| Processor | Quad-core 3.6 GHz Intel Core i7-4790 (x64) |
| Memory | 16 GB RAM |
| Storage | 1 TB HDD |
| OS | Linux |
| Power | 350 W power adapter |
| LAN | Realtek PCIe 10/100/1000 Mbps Gigabit Ethernet |
| Costs (asset/working) [$] | 1000/0.11 |

restrictions, a comprehensive, dense and economic infrastructure of cloudlets using this deployment concept becomes unrealistic.

**Wireless Home Router as Cloudlet.** We built a proof-of-concept prototype to show the feasibility and explore the performance of our concept. For that, we use a customary wireless home router (Asus RT-AC87U) with a dual-core ARM processor, 256MB memory and OpenWRT[7], an open-source linux-based operating system (cf. Table 4). This operation system provides us SSH access to the router. Taking no account of security and privacy for the first prototype, we installed required softwares directly on the router's system. We chose NodeJS[8] - an open source, lightweight, cross-platform runtime environment - for building our network application. Three main benefits were decisive: *firstly*, the fast and easily developing on high-level programming language (JavaScript). *Secondly*, NodeJS is built on C++-written *Google's V8 JavaScript engine*[9] that is extremely fast, uses minimal resources and compiles JavaScript source code

[7] https://openwrt.org (accessed 2015-08-10).
[8] https://nodejs.org (accessed 2015-08-10).
[9] https://code.google.com/p/v8 (accessed 2015-08-10).

**Table 4.** Wireless Home Router specifications

| | |
|---|---|
| Model | Asus RT-AC87U |
| Processor | Dual-core 1 GHz Broadcom BCM4709 (ARM Cortex-A9) |
| Memory | 256 MB RAM |
| Storage | 128 MB |
| OS | DD-WRT |
| Power | 19 V, 1.75 A |
| WLAN | IEEE 802.11 a/b/g/n/ac, $4 \times 4$ dual-band (2.4/5 GHz) |
| Costs (asset/working) [$] | 270/0.005 |

directly to native machine code. *Thirdly*, we can reuse and easily deploy the same code for data processing to the servers (e.g., cloud, cloudlet). Thus, NodeJS matches all requirements to build real-time networking applications. We open a socket for leveraging computational power of router via wireless technologies (802.11) by the mobile device. In our first prototype we used established Internet protocols: TCP as transport protocol and HTTP as application protocol. In this case, we are able to send and compare same requests to wireless router, cloudlet or cloud.

While a deployment of cloudlets required additional computing hardware as proposed by [24] is very expensive, our concept is based on a simple firmware update of already existing infrastructure components (i.e., wireless home router). Household owners do not have additional acquisition costs. Due to the fact that routers as network devices are already continuously online, we recognize minimal high operational costs for utilizing additional computing power. Nevertheless, for that, household owners benefit from offloading possibilities to other routers.

### 4.2 Measurement Methodology

**Application Profiler.** Program profiling is an obvious approach for optimization and comparison systems [5,31]. Thus, an implemented lightweight runtime profiler (i.e., an Android app running in the background) measures following metrics for our benchmarks: task completion time, processing time, and network delay time. In addition to them, the profiler permanently monitors and logs resource usages: CPU usage, memory usage, and energy consumption on the mobile device. We chose a sampling rate of 500 ms for CPU and memory monitoring and a sampling rate of 50 ms - a good, empirical determined balance between accuracy and CPU load - for energy measurements.

**Dataset and Computational Task.** While our main goal is to compare performance locally against offloading concepts, the choice of the dataset and the computational task is secondary and replaceable. We chose a set of sensor data, more precisely raw location values, and evaluate them for place detection utilizing resource-intensive clustering algorithm DBSCAN with an overall average

**Table 5.** Theoretical and measured network configurations

| Network | Theoretical bandwidth | Measured bandwidth (up-/download) [Mbps] |
|---|---|---|
| LAN | 100 Mbps–1Gbps | 310.80 ± 120.39 |
| WLAN (802.11n/ac) | 6.5–300 Mbps ($4 \times 4$, 20 MHz) | 160.95 ± 23.12 |
| DSL (6,000) | 6,016 kbps | 0.63 ± 0.04/5.44 ± 0.77 |
| GSM (2G) | 9.6 kbps | 0.24 ± 0.06/0.10 ± 0.01 |
| UTMS (3G) | 384 kbps | 1.90 ± 0.25/5.75 ± 1.00 |
| LTE (4G) | 150 Mbps–1 Gbps | 2.19 ± 0.28/16.00 ± 1.43 |

runtime complexity of $\mathcal{O}(n \log n)$ [11]. But other responsive use cases or dataset are imaginable, e.g., speak, activity, face, object or gesture recognition [20]. To ensure repeatability across different benchmark runs, the input data consisting of location values is fixed and equal, i.e., we ignore the tracking of sensor data that is not relevant for this paper, but we reference to our previous work for measuring sensor tracking [26]. For our benchmark purpose, we created six datasets varying in their data size (50 kB, 100 kB, 200 kB, 300 kB, 400 kB, 500 kB) in advance to measure their impact.

**Measurements.** We tested 15 different scenarios consisting of local and offloading processing: (1) locally on the device, (2) cloudlet over wireless LAN, (3–14) three different located clouds (US West, EU, Asia Pacific) over four different wireless networks (2G, 3G, 4G, wireless LAN/DSL), and (15) our router-based cloudlet concept over wireless LAN. The theoretical and measured network configuration used in our benchmark tests can be found in Table 5. For measuring, we disabled all background services not required for running the operating system. The display was switched off during the measurement runs. We start to monitor and log energy consumption, CPU and memory usage. Each measurement scenario was then measured with our six different datasets as follows: *first*, we run a baseline measurement for 30 s to get the default average resource usage of operating system processes and our profiler tool. *Second*, we executed five times the same task processing with the same dataset - either locally or on a remote system depending on the scenario - and measure for each task processing its completion time consisting of network delay and processing time on the executable system. A task processing run works as follows: the smartphone sends the specific dataset to the offloading system, the offloading system processes these data by executing the DBSCAN algorithm, and sends the resulting clusters as well as the processing time back to the mobile device. *Finally*, the resulting values of these five runs were averaged to reduce measurement errors. From these values (i.e., energy consumption, CPU and memory usage) were subtracted the baseline values to get isolated values only for the offloading tasks. In the next section, we report these benchmark results and discuss implications.

## 5 Benchmark Results

To prove our novel and economic deployment concept of cloudlets in terms of performance and being suitable for daily use, we conducted benchmark tests against local processing on the mobile device and existing state of the art offloading concepts, i.e., cloud and cloudlets. While all three clouds in our test are equipped with the same resources (cf. Table 2), we only report one of them in our benchmark results for better clarity. For that, we chose the one (US West instance) with the intermediate latency of the three various distant clouds (cf. Fig. 3).

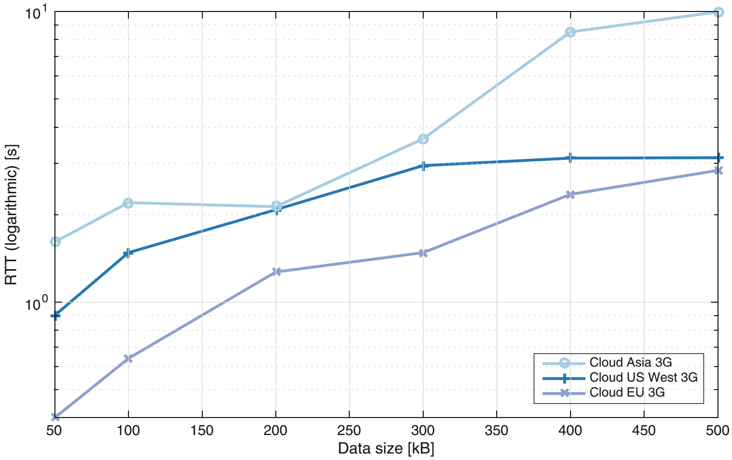
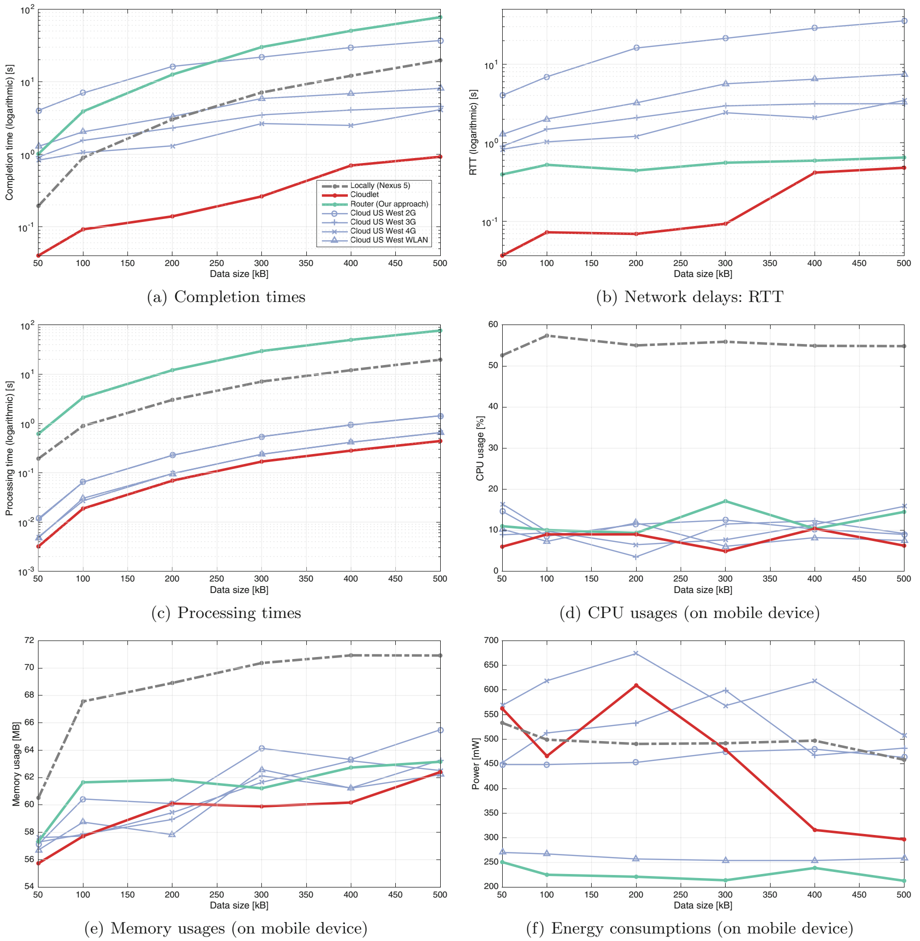



**Fig. 3.** Comparison of network delays between different located Amazon clouds

Figure 4 shows the entire benchmark results over different computational tasks, where the computational expense and the network traffic depends on the data size. We measured completion times consisting of network transmission delay (except in the case of local processing) and the pure processing time for analyzing the sensor data (cf. Fig. 4(a)–(c)) as well as resource usages on the mobile device, i.e., cpu usage, memory usage, and energy consumption (cf. Fig. 4(d)–(f)).

Considering completion times, cloudlets with additional hardware (comparable to the cloud resources in this benchmark tests) are the best choice in our computational task use case (cf. Fig. 4(a)). Our router-based cloudlet approach, that does not need any additional hardware, even outperforms the clouds with weak Internet access at small data sizes. Local processing on mobile device is sufficient at small data sizes because of enough computational power for that task and no network delay. However, if the complexity of the computational task increases, the need for offloading becomes obvious. In our laboratory test, the offloading systems are only utilized by one client. But, we need to consider




(a) Completion times

(b) Network delays: RTT

(c) Processing times

(d) CPU usages (on mobile device)

(e) Memory usages (on mobile device)

(f) Energy consumptions (on mobile device)

**Fig. 4.** Benchmark results over different computational tasks (represented by data size); where (a)–(c) are metrics to measure the offloading process while (d)–(f) monitor resources of the mobile device during that process. Our router approach is colored in *green*, cloudlet in *red*, US West cloud with four different network configurations (2G, 3G, 4G, WLAN) is displayed in *blue* and local processing on mobile device in *black* (Color figure online).

performance losses in real world scenario because of having multiple clients connecting to offloading systems and using the shared resources. The count of connecting clients strongly depends on range restrictions: while clouds are accessible from everywhere over Internet, cloudlets and wireless routers are only accessible in their radio range. As expected, cloudlets and our router-based cloudlet approach have lowest network transmission delays over all data sizes due to their nature of nearby located computing capabilities accessible over wireless LAN (cf. Fig. 4(b)). Depending on the used network technologies the network delay to the clouds increases the smaller the possible bandwidth (i.e., WLAN, cellular network: 4G, 3G, 2G). While processing times of clouds and cloudlet are almost the same due to their similar hardware resources, processing times locally on mobile device and our router-based cloudlet are constantly higher (cf. Fig. 4(c)).

While today's smartphones are quipped with relative performant hardware for their small form factor, reasons for offloading becomes directly visible by having a look at the resource usages on the mobile device. Unsurprisingly, processing tasks locally uses much more computational power than the network transmission process for offloading tasks on average (cf. Fig. 4(d)). The same is true for the average memory usage on the mobile device (cf. Fig. 4(e)). A high average utilization on the mobile device dramatically decreases the user experiences. Interestingly, our approach of router-based cloudlet outperforms local processing and other offloading systems in averaged energy consumption during the task processing, especially cloud offloading over cellular network technologies (cf. 4(f)). Offloading systems over wireless LAN connection perform the best, i.e., high energy consumers are mobile device's processor and connections over cellular network. In summary, mobile users benefit from our router-based cloudlet concept in terms of low network latency, low resource usage and particularly low energy consumption.

## 6 Discussion and Future Work

In this section, we discuss our concept considering benchmark results and give an outlook and potentials of future works.

### 6.1 Router's Performance

While the benefits of wireless routers as cloudlets are obvious (e.g., low latency, high bandwidth, economic), the performance of router is weaker than other state of the art offloading systems and even local processing. The reason for this is that typical routers are primarily constructed for routing tasks. Nevertheless, latest home routers are already equipped with multi-core and offloading processors for concurrent task processing and will become more and more powerful. We will also connect neighboring wireless routers of various households to a computing mesh network and increase both the computational power as well as the range for connecting to this infrastructure. Such dense and decentralized infrastructure is well suited for distributed computing (inspired by SETI@Home [2]) and is also resilient in disaster scenarios, as proposed and proved in our previous work [19]. Additionally, each router can also use its connected existing intranet resources (e.g., smart tv, laptop) after a software-based upgrade through (wireless) LAN to overcome performance issues. In this scenario, the router acts as master and distributes computational tasks over its dynamically online LAN resources.

### 6.2 Offloading Strategy

We see our concept of router-based cloudlets as economic complement to existing offloading systems to enable large-scale deployment. In this light, while wireless routers are always connected to Internet, we will research in offloading strategies where the router decide when and where to offload computational task, e.g., to the cloud. It is also imaginable that routers accessible through high-bandwidth WLAN preprocess specific data to reduce network traffic to distant offloading systems.

### 6.3 Discovery, Handover, and Failure Handling

*How can mobile users discover and connect to router-based cloudlets?* is still an open and important question to make cloudlets suitable for daily use. Inspired by cellular network technologies that solve some of these issues, e.g., handover of computational tasks, a failure handling strategy for the case if the mobile user gets out of range before the task is finished.

## 7 Conclusion

In this paper, we proposed a novel concept for enabling a large-scale deployment of cloudlets only using existing infrastructure by software-based upgrading wireless home routers. Beside router's native purpose of routing data packets through the network, it can now offer computing resources with low latency and high bandwidth without additional hardware.

Proving our concept in terms of performance and being suitable for daily use, we conducted benchmark tests against local processing on the mobile device and existing state of the art offloading concepts, i.e., cloud and cloudlets. As result, we cannot show computational performance gain but low network delays and traffic towards existing offloading systems by now. Nevertheless, overcoming computational weaknesses, e.g., through also utilizing connected intranet resources by software-based upgrade or building computing mesh network with neighboring wireless routers, this concept provides enormous potentials for real world usage of in-network computing capabilities.

The feasibility of this concept is already given. Router-based cloudlets provide a promising and complementary way to enable a large-scale deployment of cloudlets in existing infrastructures. This also opens an interesting field for diverse real-time constrained and contextual applications, e.g., assistance systems or face recognition.

**Acknowledgments.** This work has been co-funded by the LOEWE initiative (Hessen, Germany) within the NICER project and by the German Research Foundation (DFG) as part of project B02 within the Collaborative Research Center (CRC) 1053 – MAKI.

## References

1. Aijaz, A., Aghvami, H., Amani, M.: A survey on mobile data offloading: technical and business perspectives. IEEE Wireless Commun. **20**(2), 104–112 (2013)
2. Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: SETI@home: an experiment in public-resource computing. Commun. ACM **45**(11), 56–61 (2002)
3. Balan, R., Flinn, J., Satyanarayanan, M., Sinnamohideen, S., Yang, H.-I.: The case for cyber foraging. In: 10th Workshop on ACM SIGOPS European Workshop, pp. 87–92. ACM (2002)
4. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: 1st Workshop on Mobile Cloud Computing (MCC 2012), pp. 13–16. ACM (2012)
5. Carroll, A., Heiser, G.: An analysis of power consumption in a smartphone. In: USENIX Annual Technical Conference, vol. 14 (2010)
6. Chen, Y., Liu, B., Chen, Y., Li, A., Yang, X., Bi, J.: PacketCloud: an open platform for elastic in-network services. In: 8th International Workshop on Mobility in the Evolving Internet Architecture (MobiArch 2013), pp. 17–22. ACM (2013)
7. Chun, B.-G., Ihm, S., Maniatis, P., Naik, M., Patti, A.: CloneCloud: elastic execution between mobile device and cloud. In: 6th Conference on Computer Systems (EuroSys 2011), pp. 301–314. ACM (2011)
8. Clinch, S., Harkes, J., Friday, A., Davies, N., Satyanarayanan, M.: How close is close enough? understanding the role of cloudlets in supporting display appropriation by mobile users. In: 10th International Conference on Pervasive Computing and Communications (PerCom 2012), pp. 122–127. IEEE (2012)
9. Cuervo, E., Balasubramanian, A., Cho, D.-K., Wolman, A., Saroiu, S., Chandra, R., Bahl, P.: MAUI: making smartphones last longer with code offload. In: 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010), pp. 49–62. ACM (2010)
10. Dinh, H.T., Lee, C., Niyato, D., Wang, P.: A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing **13**(18), 1587–1611 (2013)
11. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2th International Conference on Knowledge, Discovery and Data Mining (KDD 1996), vol. 96, pp. 226–231 (1996)
12. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: a survey. Future Gener. Comput. Syst., Elsevier **29**(1), 84–106 (2013)
13. File, T.: Computer and internet use in the United States. Current Population Survey Reports, P20–568. US Census Bureau, Washington, DC (2013)
14. Keller, R., Choi, S., Dasen, M., Decasper, D., Fankhauser, G., Plattner, B.: An active router architecture for multicast video distribution. In: 19th International Conference on Computer Communications, vol. 3, pp. 1137–1146. IEEE (2000)
15. Khan, A.K., Kiah, M.L.M., Khan, S.U., Madani, S.A.: Towards secure mobile cloud computing: a survey. Future Gener. Comput. Syst., Elsevier **29**(5), 1278–1299 (2013)
16. Khan, K.A., Wang, Q., Grecos, C., Luo, C., Wang, X.: MeshCloud: integrated cloudlet and wireless mesh network for real-time applications. In: 20th International Conference on Electronics, Circuits, and Systems (ICECS 2013), pp. 317–320. IEEE (2013)

17. Makris, P., Skoutas, D.N., Skianis, C.: On networking and computing environments' integration: a novel mobile cloud resources provisioning approach. In: International Conference on Telecommunications and Multimedia, pp. 71-76. IEEE (2012)
18. Mell, P., Grance, T.: The NIST Definition of Cloud Computing (2011)
19. Panitzek, K., Schweizer, I., Schulz, A., Bönning, T., Seipel, G., Mühlhäuser, M.: Can we use your router, please?: benefits and implications of an emergency switch for wireless routers. Int. J. Inf. Syst. Crisis Response. Manage. **4**(4), 59–70 (2012)
20. Ra, M.-R., Sheth, A., Mummert, L., Pillai, P., Wetherall, D., Govindan, R.: Odessa: enabling interactive perception applications on mobile devices. In: 9th International Conference on Mobile Systems, Applications, and Services (MobiSys 2011), pp. 43–56. ACM (2011)
21. Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R.: Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun. Surv. Tutorials **16**(1), 369–392 (2014)
22. Satyanarayanan, M.: Fundamental challenges in mobile computing. In: 15th Symposium on Principles of Distributed Computing (PODC 1996), pp. 1–7. ACM (1996)
23. Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Pers. Commun. **8**(4), 10–17 (2001)
24. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)
25. Satyanarayanan, M., Lewis, G., Morris, E., Simanta, S., Boleng, J., Ha, K.: The role of cloudlets in hostile environments. IEEE Pervasive Comput. **12**(4), 40–49 (2013)
26. Schweizer, I., Bärtl, R., Schmidt, B., Kaup, F., Mühlhäuser, M.: Kraken.me mobile: the energy footprint of mobile tracking. In: 6th International Conference on Mobile Computing, Applications and Services (MobiCase 2014), pp. 82–89. IEEE (2014)
27. Seybert, H.: Internet use in households and by individuals in 2011. Eurostat Stat. Focus **66**, 2011 (2011)
28. Stojmenovic, I.: Fog computing: a cloud to the ground support for smart things and machine-to-machine networks. In: Telecommunication Networks and Applications Conference (ATNAC 2014), Australasia, pp. 117–122. IEEE (2014)
29. Tennenhouse, D.L., Smith, J.M., Sincoskie, W.D., Wetherall, D.J., Minden, G.J.: A survey of active network research. Commun. Mag. **35**(1), 80–86 (1997)
30. Verbelen, T., Simoens, P., DeTurck, F., Dhoedt, B.: Cloudlets: bringing the cloud to the mobile user. In: 3th Workshop on Mobile Cloud Computing and Services (MCS 2012), pp. 29–36. ACM (2012)
31. Wang, C., Li, Z.: A computation offloading scheme on handheld devices. J. Parallel Distrib. Comput. **64**(6), 740–746 (2004)