Analysis of "Cloudlet-based Efficient Data Collection in Wireless Body Area Networks" report

Andrea Graziani
andrea.graziani93@outlook.it
Università Degli Studi di Roma Tor Vergata
Rome, Italy

1 INTRODUCTION

1.1 Research's Goal

> *"The main goal of this paper is to develop a large scale WBANs system in the presence of cloudlet-based data collection model. The objective is to minimize end-to-end packet cost by dynamically choosing data collection to the cloud by using cloudlet based system [...] reducing packet-to-cloud energy, the proposed work also attempt to minimize the end-to-end packet delay"[9, par. 1.4]*

According to Quwaider and Jararweh [9], the goal of their work is to built an efficient **Wireless Body Area Networks (WBAN)** exploiting **edge computing**, a new paradigm in which substantial computing and storage resources, referred to as **cloudlets**, are placed at the Internet's edge, that is in close proximity to WBAN devices or sensors.[10]

To be more precise, as stated by Quwaider and Jararweh [9], edge computing resources are exploited in order to **minimize average power consumption and delay** when **Personal Digital Assistant (PDA)** transmit collected data.

1.2 Edge Computing Application Typology

> *"In our implementation, the enterprise cloud system will be the ultimate destination of the collected data. The enterprise cloud is also able to send messages back to both the cloudlet system or to WBAN users."[9, par. 3.2]*
>
> *"The enterprise cloud system is a centralized management and storage point that can be accessed by different organizations that are interested in a certain type of data. Another important feature of the cloudlet system is the ability of bidirectional communications between many WBANs users."[9, par. 4.1]*

WBAN devices and sensors have very stringent constraints in term of CPU performance, storage resources and battery life.

To perform a wide range of healthcare, military and sport application, **Tier-1**, which represents "*the cloud*" in today's parlance, plays a very important role in this kind of system: cloud resources are exploited to overcome WBAN devices constraints, since Tier-1 represents, in terms of archival preservation, the *safest place to store data*, ensuring the long-term integrity and accessibility, and the *main execution site to perform expensive computations*, thanks to its almost unlimited elasticity.

However cloud resources exploitation isn't a panacea due to some negative consequence, as the increasing of network **round-trip times (RTT)** experienced by mobile users.

Therefore, Quwaider and Jararweh [9] have built a system capable to exploit edge computing resources too, in order to offload compute-intensive operations at very low latency.[11][10]

To be more precise, WBAN system prototype modelled by Quwaider and Jararweh [9] is intended for *edge-accelerated, cloud-native applications*. Why?

cloud-native because, to overcome stringent constraints on battery, performance and storage of WBAN devices, cloud resources are exploited, through offloading techniques over a wireless network, **to execute critical task**; in other words, as stated also by Quwaider and Jararweh [9], *cloud is the primary execution site and it is essential to provide services to final users.*

edge-accelerated because the system exploits edge computing resources **only when available**. In that way, that system is capable to provide optimal performance, but *edge resources represents a secondary execution site, that is they are optional to provide services.*

According to Satyanarayanan et al. [11], it is shown that edge-accelerated, cloud-native applications can have a *modest* advantages in term of *response times*, *battery life*, *ingress bandwidth demand reduction*, *privacy* and *fallback services*[11][10], but, as Satyanarayanan et al. [11] states, edge computing potentiality is not fully exploited due to central role assigned to cloud.

However, it is true that any edge-accelerated, cloud-native applications involves less investment risk, much less software development, and their markets are much larger since they *can function acceptably even in the absence of edge computing*.

1.3 Edge Computing Resources exploitation

How edge computing resources are been exploited in order to improve, as stated by Quwaider and Jararweh [9], average power consumption and delay during packet transmissions?

1.3.1 Using an energy-efficient communication technology. First of all, is very important to precise that Quwaider and Jararweh [9] research is based on the observation that using WiFi technology a WBAN user will be able to transmit data packet to the cloud with **low power consumption**, **low delay** and mostly **no connection cost** compared with cellular technology; in fact, they wrote:

> *"It was shown that, via WiFi, the transmission power of a data packet of size 46 Bytes will cost about 30 mw and with a delay of 0.045 ms. On the other hand, a*

> *longer transmission range cellular network connection (e.g. 3G and LTE) is capable of transmitting the data packet to the cloud from any location that is cover by cellular network, which is usually a wider geographic area compared with the WiFi. It was shown that, via cellular, the transmission power of data packet of size 46 Bytes will cost about 300 mw and with a delay of 0.45 ms."[9, par. 3.1]*

Therefore, since WiFi technology is more energy-efficient than cellular technology, if WBAN users have the chance to use it, is possible to reduce average power consumption during packet transmission.

However, how to exploit WiFi technology in WBAN system taking into account its **short transmission range** compared to cellular network? Providing an edge computing infrastructure made up of several cloudlets geographically distributed and equipped with computational, storage and communication capability, increasing WiFi coverage. In fact, researchers wrote:

> *"The cloudlet system is composed of set of physical servers with many cores and huge Gigabytes of memory. The cloudlet server system is equipped with one or more of the communication antennas that is supporting different physical layer capabilities (e.g. WiFi and WiMax)."[9, par. 4.1]*

However there is a problem: since WiFi technology transmission range is short and edge infrastructure resources are expensive, cloudlet and WiFi coverage aren't available everywhere. Therefore, Quwaider and Jararweh [9] implemented a *self-adaptive behaviour* according to which **communication technology is dynamically changed** when users move from one region to another. This is the reason according to which Quwaider and Jararweh [9] modelled their system identifying three different regions:[9, par. 3.1]

Cloudlet Region (CR) where WiFi coverage is available, so a user can use it to transmit data to the cloudlet.

Enterprise Region (ER) where only cellular coverage is available, therefore a user can use only cellular technology to transmit a data packet to cloud.

Not-covered Region (NC) where neither WiFi nor cellular technology is available. In this case a user should buffer the packets until one of the above technologies is available, then to be able of transmitting the packet to the enterprise cloud.

In other words, WBAN users are capable to exploit edge computing resources if and only if they are in a *Cloudlet Region*, otherwise they go on transmitting data directly to the cloud, if possible. This is the reason according to which we have previously defined Quwaider and Jararweh [9] application as *edge-accelerated*: cloud is very critical for that application since it represents the ultimate destination of collected data and perform critical task providing access to data. In other words, seem enough clear that cloudlet is less critical compared to cloud to providing system services. In fact, Quwaider and Jararweh [9] wrote:

> *"In our implementation, the enterprise cloud system will be the ultimate destination of the collected data. The enterprise cloud is also able to send messages back to both the cloudlet system or to WBAN users."[9, par. 3.2]*

> *"The enterprise cloud system is a centralized management and storage point that can be accessed by different organizations that are interested in a certain type of data."[9, par. 4.1]*

1.3.2 Exploiting network proximity. Deploying cloudlet *close* to final users, thanks to **high-bandwidth single-hop connection**, is possible to reduce RTT and to increase end-to-end bandwidth; in other words, we can achieve **network-proximity**.[11]

However, is very important to not confuse *physical proximity* with *network-proximity*. Cloudlet physical proximity not always can affect positively RTT since it does *not* guarantee network proximity: think for example to a highly congested WiFi network which may have poor RTT. Theoretically, is possible to achieve network proximity without physical proximity, for example using a fiber link between a wireless access point and a cloudlet that is many tens or even hundreds of kilometres away assuring therefore low RTT and high bandwidth.[11]

1.3.3 Using cyber-foraging techniques. WBAN sensors and devices collect a very huge amount of data. In fact, Quwaider and Jararweh [9] wrote:

> *"The multiple WBAN sensor nodes are capable of sampling, processing, and communicating one or more vital signs like heart rate, blood pressure, oxygen saturation, breathing rate, diabetes, body temperature, ECG and activity, or environmental parameters like location, temperature, humidity, light, movement, proximity and direction."[9, par. 1.1]*

Obliviously, WBAN devices and sensors cannot have enough computation and storage resources due to their very strictly constrains in term of size and power consumption.

In order to overcome WBAN devices limitation, increasing energy efficiency, several cyber-foraging techniques are used: Quwaider and Jararweh [9] adopt a **Data Staging Tactic** to allow WBAN devices, after offloading collected data to cloudlet or cloud, to free up storage space. To achieve energy-efficiency, all computations regarding collected data are carried out by cloud or cloudlet, therefore **Computation Offload Tactic** is used too.

Quwaider and Jararweh [9] adopt a very static approach to offloading since, if a offload target is available, data will be *always* offloaded. However, WBAN users devices exhibit a self-adaptive behaviour according to which they can decide to offload towards cloud instead of clodulet if Wifi network is highly congested; in fact, researchers wrote:

> *"each VC region is able to serve a certain number of users [...] Then, the extra users within the VCs have to send the data via the cellular communication, even though, they are within the VCs."[9, par. 5.1]*

1.4 A self-adaptive system

It's clear that Quwaider and Jararweh [9] had built a **self-adaptive system** because it is capable to change its behaviour when execution environment changes. Let's now try to be more precise:

- From a deployment point of view, this system represents a **distributed self-adaptive system** since a self-adaptive software, including both *managed software* and *managing software*, is deployed in every WBAN users device.[13]
- From decisions control point of view, we can consider it a **decentralized self-adaptive system** too due to the lack of a central control component that decides about when and how to perform an adaptation; in other words, all WBAN nodes are responsible for adaptations, therefore control decision is decentralized.[13]

  However, is very important to precise that in Quwaider and Jararweh [9] system there is no coordination between WBAN nodes referring to control decision. All of them act independently, deciding adaptations autonomously, therefore, any kind of control decisions decentralization pattern is been used.
- From an architectural point of view, that system is based on a **MAPE-K feedback control loop** and represents a **autonomic system**. In fact, all WBAN nodes act as *autonomic elements* because they manage their behaviour in accordance not only with execution environmental changes, but according to *policies and goals* established by [9] too. In other words, that system has **self-management capabilities**, relieving WBAN users of the responsibility of directly managing their device changing communication technology to achieve established goals; as known, self-management is the essence of autonomic computing.[4]

  Let's study in deep the control feedback loop of that system:

  Knowledge/Goal The goals established by Quwaider and Jararweh [9] are average power consumption and average transmission delay minimization.

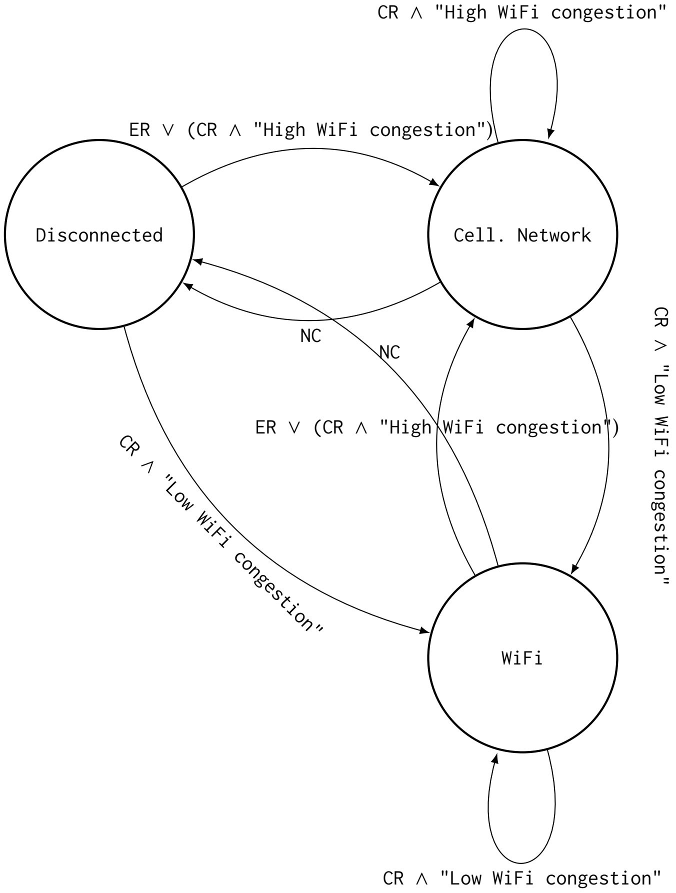
  Monitor The monitoring process is responsible for collecting data from environment like network quality and WiFi or cellular connection availability.

  Analyse Measured data are analysed in order to detect WiFi congestion condition level ("Low" or "High") and check what is the current region ("NC", "ER" or "CR") where user is.

  Plan Select and plan a solution to achieve established goal in according to system state diagram represented in 1.

  Execute Change communication technology according to current state.
- Quwaider and Jararweh [9] solution can be considered as **application-transparent** since WBAN applications are *not* informed that a communication technology change is occurred; in other words, underlying system is the only responsible for the adaptation. In this way, compatibility with already existing applications is assured, although adaptations can affect negatively on WBAN applications.

Figure 1: State diagram of communication technology used by WBAN devices


CR ∧ "High WiFi congestion"
ER ∨ (CR ∧ "High WiFi congestion")
Disconnected
Cell. Network
NC
NC
ER ∨ (CR ∧ "High WiFi congestion")
CR ∧ "Low WiFi congestion"
CR ∧ "Low WiFi congestion"
WiFi
CR ∧ "Low WiFi congestion"

1.5 Communication style

Basically, when a device cannot transfer data packet to cloudlet for some reason, WBAN users are obliged to communicate directly with server-side application running on cloud, therefore a classical **client-server style communication** is adopted, which use classic *point-to-point* and *synchronous* communications. As known, use this kind of communication can be problematic due to **high degree of coupling between server and client**, like the lack of failure transparency or latency issue due to blocking nature of that kind of communication.

However, when a WBAN user is capable to communicate through a server, the system adopts another communication style: **server-side client/agent/server model**. In that case, the *agent* is the cloudlet, which, acting as a surrogate to perform cyber-foraging tactics, is located on the so-called "*fixed*" side portion of the network and, precisely, on the edge of the network. That **loose-connector** style is capable to reduce coupling degree between server and client achievement several advantages in term of failure transparency (cloudlet can mask temporary cloud application unavailability).

2 SIMULATIONS ANALYSIS

In these section we will analyse simulation's results performed by Quwaider and Jararweh [9] in order to quantify performance advantages by using edge computing resources.

WBAN system performance is evaluated through several simulation sets focusing mainly on two performance metrics:

> *"Packet Transmission Power and Packet Delay. The Packet Transmission Power and Packet Delay are directly measure of the communication energy and delay expenditure from the Personal Digital Assistant (PDA) device to the cloudlet or the enterprise cloud"[9, par. 4.2]*

As stated by Quwaider and Jararweh [9], experiments were carried out using a custom version of *CloudSim*[1] simulator, monitoring a virtual space-area, where several cloudlet entities were deployed in known geographic locations, in such a way transmission range belonging to different cloudlet not overlapping, while some WBAN users were moved randomly with fixed speed and random pause time.

2.0.1 MAC protocol. According to Quwaider and Jararweh [9], the medium access mechanisms used in their simulation is **pooling-based**, that is a strictly centralized scheme, also called **point coordination function** (**PCF**), in which the access point, called **master**, dynamically polls clients for data. According to this scheme, the master is allowed to build a list of stations wishing to transmit during a contention phase. After this phase, the station polls each station on the list.

In a WBAN's context, this choice has several advantages:

- PCF is capable to offer time-bounded service, **guaranteeing a maximum access delay and minimum transmission bandwidth** making the system more **predictable** and, therefore, more suitable for **real-time** health monitoring; in fact, if a predictable system is used, will be possible to know with reasonable approximation the *worst case execution time* (WCET) of jobs and check if any hard real-time constraints is respected.
- Since WBAN users are not allowed to send data without the master's invitation, the **hidden terminal problem is eliminated**. Keep in mind that using DCF (Distributed coordination function) with RTS/CTS extension, in order to resolve hidden terminal problem, is not suitable in a WBAN's context because to short frame (only 46 byte) send by devices; RTS/CTS extension would introduce a non-negligible overhead causing a waste of bandwidth, higher delay and energy consumption.[12]
- This scheme allow higher throughput due to less collisions respect to CSMA/CA.[12]

Obliviously this choice isn't a panacea; it can introduce higher delay under a light load and overhead if nodes have nothing to send.[12]

2.1 First experiment set

First experiment set goal is to quantify both **packet process delay** and **power consumption**, due to packet computation, by cloudlet varying following system parameter.

(1) number of virtual machine deployed on a cloudlet (0, 2, 4 or 8). The maximum number of deployable virtual machines is bounded by the number of physical processors available on cloudlet (Quwaider and Jararweh [9] prototype cloudlet have 8 physical processors, therefore 8 is the maximum number of deployable VMs).
(2) processing speed of data packet, from a minimum 100 to a maximum of 900 **million instructions per second** (MIPS).
(3) number of WBAN users (up to 150 users).

According to experiments results, [9] have shown that:

(1) Increasing the number of virtual machines running on a cloudlet, packet process delay will be reduced.
It is shown that, if data packets can be processed in **parallel**, the **blocked time** can be reduced. In fact, fixed data packet *average arrival rate* λ and *average service rate* μ of each virtual machine, system cloudlet **utilization**, that is *the fraction of time according to which cloudlet is busy*, is reduced; each virtual machine, by symmetry, sees an arrival rate of $\frac{\lambda}{k}$, where k is the number of deployed virtual machines. Therefore cloudlet utilization, which is equal to $\frac{\lambda}{k\mu}$, decrease increasing k. It is shown that, decreasing utilization, **blocked time** decreases.[1]
From an architectural point of view, the increase of the number of virtual machines on a cloudlet can be considered as a **performance tactic** and, precisely, a **resource management tactic**, because affects the time within which a response is generated with a better resources management achieved through concurrency.[1]

> *"the processing time is decreased by approximately 85% by using cloudlet system configured with 8 VMs comparing to using only one VM in CS."[9, par. 3.3]*

(2) Fixed the average time required to process a packet data on a CPU, that is for a given MIPS process speed, power consumption and processing delay are increased by increasing the number of users. This happens because cloudlet utilization increases because λ is higher, therefore the fraction of time according to which cloudlet is busy increases too; so, since more time is needed to complete user tasks, more power are consumed.

2.2 Second experiment set

Second experiment set was carried out in order to quantify advantages of using edge computing resources. To be more precise, that simulation monitors the effects on **average transmission power and delay** of data packed send by users PDA to cloudlet, varying following system parameters:

(1) number of cloudlet deployed (from 0 to 6).

[1] http://www.cloudbus.org/cloudsim/

(2) WBAN user's positions (speed of 2 m/s and a random pause time of $1 - 10$ s)

Monitored area size is fixed to 600×400 m while the total number of users is fixed to 400.

According to experiments results, [9] have shown that:

(1) As expected, increasing the number of available cloudlets in monitored area, average transmission delay of data packed is reduced. It happens since the probability that an user is close to one of them increases. Having more opportunities to transmit data within cloudlet coverage, users can benefit of cloudlet physical proximity which, as already said in previous section, *can* decrease RTT, affecting positively latency, bandwidth and jitter. [10].
As known, when cloudlet coverage is available, data is offloaded to a cloudlet using a high bandwidth single-hop connection, which decrease RTT. Conversely, when cloudlet coverage isn't available, data is offloaded to cloud using a **multi-hop network connections** which involves high RTT and likely low bandwidth connection.[5]
(2) Increasing the number of available cloudlets, average transmission power of data packed is reduced too. It is due to the use of WiFi technology which is, as already said, more energy-efficient compared to cellular network.
(3) Increasing the number of users, both average transmission power and delay increase. This happens because:
  (a) When a cloudlet area contains a large number of users, interferences increase, leading to higher error bit rate and affecting negatively both transmission delay, due to data packets or acknowledgements loss, and power, due to packet retransmission.
  (b) Since a polling MAC scheme is used, an higher number of users within an cloudlet WiFi coverage area increases the time needed to pool every node by AP.
  (c) As stated by Quwaider and Jararweh [9], an high congested WiFi network can cause WBAN devices to perform a self-adaptation action according to which communication technology is switched, using cellular network to send packet data instead WiFi, affecting negatively aforementioned metrics.

2.3 Third experiment set

The last experiment set is focusing on monitoring aforementioned performance metric varying cloudlet geographical placement. Monitoring an 800×800 m area, simulations were carried out varying following system parameters:

(1) number of cloudlet deployed (up to 16 cloudlets).
(2) number of WBAN users (up to 1400 users).
(3) WBAN user's positions (same parameters as before)
(4) cloudlet geographical placement (using very different patterns, classified in three categories by Quwaider and Jararweh [9]: *Adjacent*, *Distant* and *Intermediate*)

Experiments results are the following:

(1) As expected, independently from cloudlet deployment pattern, increasing the number of cloudlet, the impact of cloudlet geographical placement on average transmission power and delay is negligible since, in that way, the opportunities to send the data packet to a cloudlet, using WiFi and with minimum cost of power and delay, increase.
(2) Fixed cloudlet and users number, deploying cloudlet using an intermediate category pattern, that is placing cloudlet neither too far apart nor too close, system performance are better than other patterns belonging to other categories.

3 FURTHER ARCHITECTURAL CONSIDERATIONS, ISSUES AND POSSIBLE IMPROVEMENTS

3.1 The cloudlet discovery and provisioning issue

As said in previous section, Quwaider and Jararweh [9] overcome to WBAN devices limits, in term of computing power and storage, through cyber-foraging techniques like computation offload and data staging. We believe that the system model built by Quwaider and Jararweh [9] is too simple to achieve research's goals.

It is shown [6] that cyber-foraging systems have *at a minimum* the following combination of functional requirements:

- A need for computation offload, data staging, or both
- A need to provision a surrogate with the offloaded computation or data staging capabilities
- A need for the mobile device to locate a surrogate at runtime

Although WBAN users *need* to be able to locate available cloudlet in an area where stage data, **cloudlet discovery** issue has *not* been addressed by Quwaider and Jararweh [9] in any way; we believe that it is a very big mistake because cloudlet discovery affects negatively energy consumptions and response time and, anyway, it is a **functional requirements** for any cyber-foraging systems.

However there is another reason which that makes cloudlet discovery issue so important: **security**. Quwaider and Jararweh [9] wrote:

> *"On the other hand, since WBANs forward useful and life-critical information to the cloud, which may operate in distributed and hostile environments, novel security mechanisms are required to prevent malicious interactions to the storage infrastructure. Both the cloud providers and the users must take strong security measures to protect the storage infrastructure."[9, par. 3.3]*

Quwaider and Jararweh [9] should have known that addressing cloudlet discovery issue, they could have improved system security. For example, using **Cloud Surrogate Directory** tactic, according to which mobile device contacts a cloud server that maintains a list of potential cloudlet/surrogates, security is highly increased because the mobile device only needs to trust the cloud surrogate directory server and can pre-exchange credentials for authorization.

Generally, when you want to offload data or computation to a cloudlet, a selection algorithm to select best cloudlet is run. In WBAN systems, maximize energy efficiency, in order to preserve devices lifetime, is critical, therefore is preferable not run selection algorithm on WBAN devices, which can decrease energy efficiency (depending on the complexity of the algorithm and the number of monitored variables). Probably, using cloud surrogate directory tactic, according to which selection algorithms run on the cloud, is possible to improve WBAN lifetime. [6]

In the same way, **surrogate provisioning** is another functional requirements for any cyber-foraging systems which is not addressed by [9], therefore is not clear how cloudlets manage offloaded computation and/or data processing operations.

During the first experiment set, Quwaider and Jararweh [9] had shown that packet process delay is affected by the number of virtual machine deployed in cloudlet. However, provisioning tactic affects this process delay too. It is shown that Pre-provisioned cloudlet have the advantage of shorter provisioning times because the capabilities already reside on cloudlet, providing shorter response times to requests from mobile devices.[6]

3.2 IEEE 802.11ah as communication standard

In addition to changing communication technology, we believe that the use of a communication standard capable to minimize the main sources of energy waste like *collision*, *inter-network interference*, *idle listening* and *control packet overhead* is extremely useful to maximizing the lifetime of WBAN devices.

Unfortunately, Quwaider and Jararweh [9] give us very few details about MAC protocol, however is reasonable to assume that they have used legacy IEEE 802.11 protocol with 2.4 or 5 GHz bands.

We believe that we can achieve better communication performances over longer distances among a large number of low-power devices exploiting **IEEE 802.11ah**. In fact:

- The 802.11ah standard enables single-hop communication over distances up to 1000 m, utilizing sub-1 GHz license-exempt bands to provide better propagation characteristics in outdoor scenarios, like WBAN, than legacy WiFi. On the other side
- IEEE 802.11ah MAC layer improves power efficiency through frame shortening techniques, reducing overhead caused by short packets transmission which are very common in WBAN scenario.
- Several innovative concepts such as hierarchical *Association IDentification* (AID), *Restricted Access Window* (RAW), *Group Sectorization* (GS), decrease collision probability in networks with thousands of stations, resolve hidden terminal problems, support a large number of associated stations, improving scalability and power efficiency.

3.3 Resource demand tactics

In order to reduce system utilization, therefore reducing average packet process delay, we can adopt a **performance tactic** according to which the number of packet to process can be reduced *managing the packets data generation rate.*In other words, if it is possible to reduce the sampling frequency at which environmental and body variables are monitored, demand can be reduced.[1]

A possible solution suitable for WBAN could be to implement a self-adaptive behaviour according to which sampling frequency can be temporary reduced when users population within a cloudlet region grows beyond a certain threshold (other system parameters, like network conditions or residual battery life, can be considered during adaptation decision).

3.4 Variable data fidelity tactic

Unfortunately Quwaider and Jararweh [9] *didn't introduce any variable data fidelity tactic* capable to improve WBAN battery life. We believe that the introduction of an **approximate computing** (and **storage**) technique can be useful in WBAN context.

As known, approximate computing is a new paradigm based on the intuitive observation that, while performing exact computation require high amount of resources, allowing selective approximation or imperfect computations results can provide drastic energy savings.

To be more precise, approximate computing leverage the presence of error-tolerant code regions in applications and perceptual limitations of users to intelligently trade off implementation, storage and/or result accuracy for performance or energy gains. In brief, approximate computing exploits the gap between the level of accuracy required by the applications/users and that provided by the computing system.

For example, can be useful, in order to improve energy-efficiency, to reduce data fidelity collected by WBAN devices according to network condition, current position or battery status. For instance, we can adopt a technique according to which, when network condition are very bad or totally absent, in order to reduce the energy cost due to data buffering, critical application data (for example vital signs like heart rate, blood pressure, oxygen etc.) are stored in reliable memory segments with higher refresh rate, conversely less critical data (for example experimental data like location, temperature, humidity, light etc.) are stored into less reliable memory with a less refresh rate. As known, higher is refresh rate, higher is power consumption.[8]

Moreau et al. [7] had shown that, using approximate computing techniques, applications run 3.78 times faster while energy consumptions are reduced 2.77 times. Felzmann et al. [3] achieve similar results, claiming that approximate computing can show up to 30% energy savings.

3.5 Scalability and elasticity tactic

Solution proposed by Quwaider and Jararweh [9] doesn't exploit any scalability/elasticity tactic in order to increase cloudlet resource efficiency achieving scalability and elasticity. Keep running several virtual machines on a cloudlet while there are few WBAN users within a cloudlet region is a real waste of resources. Moreover full virtualization approaches are usually unsuitable for deploying distributed applications due to their overhead and long boot time.

OS-level virtualization based tools, also called **containers**, are often a better choice for scalability and elasticity purposes since they represent a lightweight alternative to virtual machine thanks to their very small footprint and quick boot time.

Exploiting container advantages and using a scalability/elasticity tactic like **Just-in-Time containers tactic**, typically used to complement the computation offload tactic and data staging tactics, we can achieve aforementioned objectives. This tactic, in order to decrease the load on a cloudlet, supporting a greater number of requests, creates an instance of the offloaded code upon receipt of an offload request, destroying it when request is fulfilled. However, because the instance is created at runtime, there is a response/execution time penalty to create the instance before computation can execute.[6][5]

3.6 Mobility issue

Quwaider and Jararweh [9] don't specify which transportation protocol is been used in their simulations and, once again, we believe that the lack of that information could be a problem because it is shown that transportation protocol has negative effects on performance in wireless communications, where many data packet can be lost due both to an *high error bit rate* and to an *high amount of disconnections* due to user mobility.

For instance, TCP interpreters packet loss as a congestion symptom and the activation of its congestion control mechanisms can, as known, negatively affect performance.[2]

Since WBAN user can move from a region to another, several hand-offs can occur, causing data packets loss and performance worsening. Unfortunately, any information is given about *average residence time* within a region by a WBAN user and the amount of hand-off performed during simulations. According to Quwaider and Jararweh [9], WiFi coverage is limited to 100 m, very little if compared to cellular network transmission range. We believe that this situation could potentially lead to frequent hand-off and low residence time affecting negativity real applications.

Quwaider and Jararweh [9] doesn't report how *users movements speed* (fixed to $2m/s$) and the *random pause time* (fixed to $1 - 10s$) affected simulation's results and is not clear how they are intercorrelated with average power consumption and delay during the second and the third experiment set. Obliviously, we will expect that user speed is correlated to the amount of hand-off, therefore faster users involve more hand-offs, however this aspect was not considered. Moreover, we believe that **cloudlet region size** is a very important parameter since smaller cloudlet regions can potentially increase the amount of hand-off performed by an user.[2] This aspect was also not considered.

Maybe, a better analysis regarding about aforementioned aspects, with a highlight system state variable describing how the state variables are interrelated, including algorithms for computing their interaction and evolution in time, would have been useful for reproducibility purposes, in order to validate Quwaider and Jararweh [9] results.

REFERENCES

[1] Len Bass, Paul Clements, and Rick Kazman. 2003. *Software Architecture in Practice (2nd Edition).* Addison-Wesley Professional.

[2] Hala Elaarag. 2002. Improving TCP Performance over Mobile Networks. *ACM Comput. Surv.* 34 (09 2002), 357–374. https://doi.org/10.1145/568522.568524

[3] I. Felzmann, J. Fabrício Filho, R. Azevedo, and L. Wanner. 2018. Impact of Memory Approximation on Energy Efficiency. In *2018 Symposium on High Performance Computing Systems (WSCAD).* 53–60.

[4] J. O. Kephart and D. M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50.

[5] Grace Lewis and Patricia Lago. 2015. Architectural Tactics for Cyber-Foraging: Results of a Systematic Literature Review. *Journal of Systems and Software* 107 (06 2015). https://doi.org/10.1016/j.jss.2015.06.005

[6] Grace Lewis, Patricia Lago, and Paris Avgeriou. 2016. A Decision Model for Cyber-Foraging Systems. https://doi.org/10.1109/WICSA.2016.38

[7] T. Moreau, A. Sampson, and L. Ceze. 2015. Approximate Computing: Making Mobile Systems More Efficient. *IEEE Pervasive Computing* 14, 2 (2015), 9–13.

[8] Veljko Pejovic. 2019. Towards Approximate Mobile Computing.

[9] Muhannad Quwaider and Yaser Jararweh. 2015. Cloudlet-based Efficient Data Collection in Wireless Body Area Networks. *Simulation Modelling Practice and Theory* (01 2015). https://doi.org/10.1016/j.simpat.2014.06.015

[10] Mahadev Satyanarayanan. 2017. The Emergence of Edge Computing. *Computer* 50 (01 2017), 30–39. https://doi.org/10.1109/MC.2017.9

[11] Mahadev Satyanarayanan, Guenter Klas, Marco Silva, and Simone Mangiante. 2019. The Seminal Role of Edge-Native Applications. 33–40. https://doi.org/10.1109/EDGE.2019.00022

[12] J.H. Schiller. 2003. *Mobile Communications.* Addison-Wesley. https://books.google.it/books?id=FdojEVT1Oj4C

[13] Danny Weyns, Bradley Schmerl, Vincenzo Grassi, Sam Malek, Raffaela Mirandola, Christian Prehofer, Jochen Wuttke, Jesper Andersson, Holger Giese, and Karl M. Göschka. 2013. *On Patterns for Decentralized Control in Self-Adaptive Systems.* Springer Berlin Heidelberg, Berlin, Heidelberg, 76–107. https://doi.org/10.1007/978-3-642-35813-5_4