MEASURE ENERGY CONSUMPTION

AIM:

Experiments relating to energy measurement could be at various levels: the hardware level; energy efficiency directive level (Simunic, et al. 2000); operating system (Sagahyroon, 2006); software application or data and user levels (Ravi, et al. 2008). Energy conservation is made possible through the use of different techniques which estimate or forecast energy consumption at the device and application level (Krintz, et al. 2004). The goal of green computing technology is to reduce carbon emission, maximize performance and prolong the lifespan of the computing resources.

this paper is to discuss the results for several investigations conducted on the energy (andbattery) consumption for running web-based and standalone applications on Windows and IOS portable computing devices. The following objectives will help to achieve this aim:

- Research Objective 1: To conduct experiments on the measurement of energy consumed for running youtube videos in different web browsers (e.g. Google Chrome, Mozilla Firefox, etc…) on Windows (i.e. laptops), and IOS machines (i.e. tablet);
- Research Objective 2: To conduct experiments on the measurement of energy consumed for playing audio and video files on several media players for windows (on a laptop);
- Research Objective 3: To conduct analyses on data collected in Research Objectives 1 and 2.

WORKING PRINCIPLE:

Electricity meters operate by continuously measuring the instantaneous voltage (volts) and current (amperes) to give energy used (in joules, kilowatt-hours etc.). Meters for smaller services (such as small residential customers) can be connected directly in-line between source and customer.
What is the principle of energy consumption?

There is a tight bond between wealth and the primary commodities that give us things like light, heat and mobility; I call it the First Principle of Energy Consumption. It simply says that the more money a person makes, the more energy they use. The First Principle works in reverse too.

This paper will be organised into the following sections: Introduction; Literature Review; Methodology; Results and Discussion; and Conclusion.

2 Literature Review

The Smart2020 report (The Climate Group and GeSI, 2008) predicts an increasing trend of BAU CO2 emissions for the ICT industry. The emissions growth rate for three ICT categories (end-user devices, telecommunication and networks, and data centers) is expected to decrease from 6.1% 3.8%. By 2020, the ICT industry's footprint is expected to rise to 1.3 $GtCO_2e$ (equivalent to 2.3% of global emissions by 2020). The PC (e.g. desktops, laptops, etc.) footprint (due to its embodied and usage emissions) is

the highest (60%) followed by printers (18%), peripherals (13%), smartphones (10%), and tablets (1%). It is estimated that the footprint of end-user devices will grow at 2.3 percent per year to reach 0.67 $GtCO_2e$ in 2020 and thus, energy efficiency improvements in these devices and their proper usage are essential for reducing their overall footprint.

2.1 Energy Consumption of software

Green and sustainable software is a software product that has the smallest possible economic, societal, ecological impact as well as impact on human beings (Ahmed, et al., 2014). This has led to the introduction of various programmes and initiatives that encourages energy efficient software such as green software engineering and Eco-design software (Kaliterre, n.d.).

According to the Greenhouse Gas Protocol (2012), applications are executed with an OS. They affect the power consumption of a device due to data requests and processing. Managing energy requires accurate measurement of the energy available and consumed by a system. This involves monitoring or estimating the resource and energy consumption of hardware and software (Noureddine, et al., 2013). However, a device's power consumption is subjected to the type of application and the task being performed which is evident in our experimental results presented in Section 4 of this paper. In order to reduce the overall power consumption for a web-based or standalone task, it will be necessary to provide users with an insight of the power consumption of the different web-based browser applications (e.g. Google Chrome, Internet Explorer, Mozilla Firefox, Safari, etc…) and also the resource hungry nature of many applications such as movie player and games.

2.2 Energy Consumption of Media Players

Modern technologies incorporate a number of power management features to reduce power waste. Dynamic Voltage and Frequency Scaling (DVFS) can enable the CPU speed to be dynamically varied based on the workload which leads to a reduced power consumption during periods of low utilization (Liu, et al., 2008). The energy-aware dynamic voltage scaling technique has been used to reduce energy consumption in portable media players (Yang & Song, 2009). This scheme showed a relationship between frame size and decoding time. These two cited work merely discuss how energy consumption can be reduced using various techniques, but have not measured the actual amount of energy being consumed by the application. However, the energy consumption of Windows Media Player has been measured using the EEcoMark v2 tool (EecoMark, 2011) but the empirical details of the measurement have not been explicitly discussed. Media playback application power consumption has been analysed by Sabharwal (2011) using windows event tracing. Event tracing does not seem to be an appropriate method for measuring energy consumption because the process itself may have impact on the results. A comparative analysis of energy consumption of media players has been conducted by Techradar (2010). The energy consumption is monitored by playing a DVD on Windows Media Player (WMP) and VLC Media Player. Their research results show that the VLC Media Player is more energy efficient than Windows Media Player. However, the cited work has not mentioned which tool has been used for measurement and additionally, the experiment procedures have not been explitcitly discussed.

Experiment Set 2: To investigate the energy consumption of several web browser applications in IOS

Experimental Steps

i. Record the battery status (in %) of the device manually;
ii. Run a 30 minutes youtube video in the following web browsers: Google Chrome, Safari, Opera Mini and Puffin;
iii. Record the battery status (in %) of the device manually immediately after (ii);
iv. Repeat all the above steps for three different times of the day: morning; noon; and night;

Constants in the experiments

i. The experiments are run in the same physical environment;
ii. The experiments are run in the same day;
iii. Volume of the device is set to *full* while the brightness, *auto*.

Critique of the experiments

i. The number of experiments for each web browser during a certain part of the day ought to be repeated at least 9 times in order to obtain an average of the readings;
ii. The tablet ought to be fully charged (i.e. 100%) before each experiment is conducted.

Experiment Set 3: To investigate the energy consumption of several media players for Windows

Experimental Steps

i. Create a csv file for saving the real time power consumption data via Joulemeter (by clicking on the *browse button*);
ii. Click on the *start saving button*;
iii. Click on the *start button* to run an audio file using KMPlayer in Windows;
iv. Click on the *stop saving button* to end the application;
v. Repeat the above (i-iv) for a video file;
vi. Repeat all the above (i-v) for the following media players: Windows Media Player (WMP) and VLC Media Player.

Constants in the experiments

i. The device's default settings are used except for the volume which is set to *100%* while the brightness of the monitor, *auto*.

Critique of the experiments

i. The number of experiments for each media player ought to be at least 4 times in order to obtain an average of the readings.

4. Results and Discussion

This section will discuss the results of the data analysis for the three sets of experiments discussed above. The Joulemeter monitored raw data is for the time stamp (in ms), power consumption (in Watts) for each component: CPU, monitor, disk, base and the application. The formula used to calculate the energy consumption by each component is: Energy (J) = Power (W) x Time (s). The results of the calculation for all the experiments runs are shown in Table 2. Note that the data is cleansed so as to omit records with application power consumption = 0W. If the number of remaining records > 50% of the raw data then the cleansed readings of the csv file will be included inthe data analysis. However, if it is otherwise, then the experiment is considered an error (see Table 2).

As previously mentioned in Section 3, an experimental error may arise due to human error. The expected total uninterrupted application running time is 2 minutes and 14 seconds (134 seconds). The aggregated expected total application time has been calculated by taking into consideration the proportion of the cleansed data and also the number experiments that have been rendered as errors. The percentages of experimental error for the 4 web browsers are shown in Table 6 and it seems that the experimental error for the Google Chrome is the lowest while the rest is approximately 5 times of Google Chrome. In order to re-affirm the validity of the findings previously discussed, it will be essential to in crease the number of experiments for each browser and measurestaken to reduce human inconsistency.

	Time in Joulemeter (s)	Expected Total application time (s)	Experimental error (%)
Google Chrome 1.3.27.5	1221.10	1206.00	1.25
Internet Explorer 9*	1096.79	1035.84	5.88
Mozilla Firefox 27.0.1	1414.85	1340.00	5.59
Safari 5.7.1	1413.87	1340.00	5.51

Table 6: Experimental Error

4.2 Web Browser Applications for IOS

Table 7 depicts the results of running experiments on an Apple ipad Air2 with and IOS operating system. A youtube video has been chosen for this set of experiments and the running time is 30 minutes. Additionally, the experiments are run forthree different times of the day: morning; noon; and night.

Table 7: Results of battery consumption for the various web browsers on an IOS device

Firstly, it is noted that the battery consumption by the Puffin web browser is consistently the highest at any part of the day. However, the findings for the rest of the three web browsers (in Table 7) are rather inconclusive. Consequently, the average battery consumption for each web browser has been calculated and shown in Table 8. Once again, the average value for Puffin seems to be the highest while it is the lowest for Google Chrome. However, this finding does not seem to be aligned to the findings in the previous section where the energy consumption for Google Chrome is higher than

Safari. In summary, in order to yield more valid results, the following measures (which have been previously mentioned) will have to be taken: conducted repeated experiments for each web browser (at least 9 times in order to obtain the avaerage battery consumption value); use a fully charged battery for each experiment.

Table 8: Average battery consumption for the various web browsers

Browser	Battery Consumption (%)				Ratio Compared to Chrome
	Morning (%)	Noon (%)	Night (%)	Average (%)	
Chrome	7.00	3.00	4.00	4.67	1.00
Safari	6.00	9.00	4.00	6.33	1.36
Opera mini	5.00	9.00	4.00	6.00	1.29
Puffin	10.00	11.00	6.00	9.00	1.93

4.3 Media Players for Windows

To reiterate, the Joulemeter monitored raw data is for the time stamp (in ms), power consumption (in Watts) for each component: CPU, monitor, disk, base and the application. The formula used to calculate the energy consumption by each component is: Energy (J) = Power (W) x Time (s). The results of the calculation for all the experiments runs are shown in Table 9. Normalised data for t = 1s is depicted in Table 10 in order to provide a fair comparison between the various media players.

Audio	Hardware Energy Consumption (J)					Application (J)	Total Energy Consumption (J)	Time (s)
	CPU (J)	Monitor (J)	Disk (J)	Base (J)	Total Hardware Energy Cnsumption (J)			
KMPlayer	1312.99	4593.31	8.38	33228.59	39143.26	433.60	39576.86	2215.24
VLC	1694.42	2870.96	11.36	37110.92	41687.66	1612.72	43300.38	2474.06
WMP	1323.99	15806.66	10.34	37046.85	54187.84	1257.62	55445.45	2469.79
Video								
KMPlayer	2387.37	29492.64	41.48	55298.70	87220.19	2305.68	89525.88	3686.58
VLC	2347.85	29278.87	8.01	54897.88	86532.61	2264.76	88797.37	3659.86
WMP	1906.01	26106.89	161.64	48950.41	77124.95	1124.46	78249.40	3263.36

Table 9: Total hardware and application energy consumption for running several media players in Windows

Audio (t=1s)	CPU (J/s)	Monitor (J/s)	Disk (J/s)	Base (J/s)	Total Hardware Energy Cnsumption (J/s)	Application (J/s)	Total Energy Consumption for t=1s (J)
KMPlayer	0.59	2.07	0.00	15.00	17.67	0.20	17.87
VLC	0.68	1.16	0.00	15.00	16.85	0.65	17.50
WMP	0.54	6.40	0.00	15.00	21.94	0.51	22.45
Video (t=1s)	CPU (J/s)	Monitor (J/s)	Disk (J/s)	Base (J/s)	Total Hardware Energy Cnsumption (J/s)	Application (J/s)	Total Energy Consumption for t=1s (J)
KMPlayer	0.65	8.00	0.01	15.00	23.66	0.63	24.28
VLC	0.64	8.00	0.00	15.00	23.64	0.62	24.26
WMP	0.58	8.00	0.05	15.00	23.63	0.34	23.98

Table 10: Normalised hardware and application energy consumption for running several media players in Windows (for t=1s)