TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 4, August 2020, pp. 1809~1818
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i4.13309

Internet-offline solution: detail description and benchmarking

Onno W. Purbo
IBI Darmajaya (Graduate Study, Computer Science, IBI Darmajaya), Indonesia

Article Info

Article history:

Received Jun 12, 2019
Revised Mar 29, 2020
Accepted April 9, 2020

Keywords:

Internet for education
Internet-offline
Linux
Open source
Rural education

ABSTRACT

In this paper, for the first time, the detail of the internet-offline solution for rural/village schools is described, and its benchmarking is reported. For providing offline knowledge access, four (4) different systems have examined. An affordable, the Raspberry Pi3 server found to be able to provide web as well as file sharing service for up to 20 clients at 100 Mbps throughput. UnixBench shows that the Raspberry Pi3 about 25% performance of a four-core i5 system. Using consumer grade power banks at 10000-20000 mAh, the Raspberry Pi3 server may provide a 6+ hour operation. With reliable electrical power, a mini PC provides a more reliable alternative at an order of magnitude increase in cost. For more than 100 users, one may use i5 or higher engines to keep the price-performance ratio low. Besides, in the internet-offline system, teachers no longer have to worry about bullies, hoax, or pornographic contents.

Corresponding Author:

Onno W. Purbo,
IBI Darmajaya (Graduate Study, Computer Science, IBI Darmajaya),
93 A.Z. Pagar Alam St., Labuhan Batu, Bandar Lampung, Indonesia.
Email: onno@indo.net.id

1. CONSEQUENCES OF ICT CLASS RE-INCLUSION INTO SCHOOL CURRICULUM

End of 2018, the Indonesian Ministry of Education interested in re-include ICT as part of the school curriculum. The consequences to considered are (1) prepare a good Informatics curriculum. This includes topics on computational thinking, basic graphical programming, robotics, 3D printing, (2) prepare the required 500,000 new ICT teachers for 240,000 schools in a short time, and (3) prepare infrastructure, laboratories, networks and must be low cost, low power consumption as electrical power is scarce in rural. Moreover, no less critical can operate without any connection to the Internet.

This work will not focus on the curriculum and educational policy this work focus on providing the infrastructure solution, especially for the village and rural schools. The Internet is scared in Indonesian rural [1]. The signal scarcity in Indonesian rural is apparent and may easily be obtained from OpenSignal apps crowdsourcing [2-4]. Consequently, for rural applications, all materials, such as multimedia files video, audio, photos as well as Internet web content including Wikipedia, should be copied to a low-cost local server called internet-offline server [5].

Internet-offline server is an open solution and well documented at OnnoCenter's wiki [5]. The primary device is a Raspberry Pi3 Server, under US$100 per unit, which may provide a small local Wi-Fi hotspot [6-8]; thus, no need for additional Wi-Fi access point in a small network. Also, any schools may easily replicate the internet-offline system. Power banks may power the whole system. Students and teachers may use smartphones as a client device [5]. Financial and budget constraints in correlation with operating and maintenance costs may require more in-depth analysis.

2. INTERNET-OFFLINE SOLUTION FOR RURAL/VILLAGES

Efforts to create internet-offline access may not be new, since the early development of the Internet, some have been looking for a solution on how to read Internet content while offline. Some may fulfill by backing up emails, SMS, WA, to more complex activities such as downloading songs, movies, files, PDFs, ebooks. Not many patents described internet-offline technology. Most patents concentrated on system and methods for providing offline viewing experience of online website content [9-11]. This work is not using any of the above patents. This work more focuses on providing the library/Internet content offline rather than viewing experience.

In this work, the focus is on providing offline access to educational content. Only a handful of institutions/researches in the world are trying to provide solutions for areas with limited Internet access. Thus, there are some activities on internet-offline initiatives. Some of the well documented and similar initiatives are SolarSPELL, WiderNet eGranary Digital Library, and Kiwix. A glimpse of these initiatives are as follows,

- SolarSPELL (solar powered educational learning library) is a student-centered initiative hosted at Arizona State University. It is a library powered by Raspberry Pi, with the WiFi access point. Today, 272 SolarSPELL digital libraries have been deployed in Fiji, Vanuatu, Samoa, Tonga, the federated states of Micronesia, Rwanda, South Sudan, and Comoros. Unfortunately, no detail circuity and schematics available [12-15].
- WiderNet Project has been around for more than 20 years. The eGranary Digital Library solution-with permission, thousands of educational Web sites has been copied and deliver them to intranet Web servers inside Widernet partner institutions in developing countries and other under-served places around the globe. Over 2,000 institutions lack adequate Internet access to adopt the eGranary Digital Library, also known as "The Internet in a Box" [16-19].
- Kiwix provides free software to bring free knowledge even to remote places, such as schools in the countryside in a developing country. Kiwix gives access to Wikipedia, WikiVoyage, Project Gutenberg, and a lot more free content from the internet-even there is no Internet connection. Kiwix is available for Android, iOS, Windows, MacOS, and Linux [20, 21].

Also, there are several initiatives mostly in Africa, both non-governmental organizations (NGOs) and state institutions. Information on the projects is available online. However, not much info on detail circuitry, schematics, apps usage, and not well documented in refereed journal articles. Some of these initiatives are,

- ICT Centre Uganda https://sites.google.com/site/ictcentreuganda/ - The ICT Center of Uganda strives to create a system so that schools can access content on the internet-offline. It seems that there are several similar efforts, especially in Africa, to create/copy Internet content so that it can be accessed offline
- EduAir-EduAir, formerly Kwiizi, from Cameroon is the concept name for offering a better education via digital with or without the Internet. Their work focuses on the design of portable and open media libraries in the form of Boxes with solar energy giving access to millions of educational content and offering an integrated communication system where learners can make video calls within the local network deployed by the Box http://www.eduairbox.com.
- Project Tawasol Tunisia - IEEE Sight in Tunisia developed Raspberry Pi operated devices with a hard disk that can be updated periodically with relevant content such as Wikipedia pages, TED Talks, and other educational content from the internet. They are capable of automatically updating content when connected to WiFi or 3G networks. No detailed configuration is available.

SolarSPELL and WiderNet Project are providing electronic library accessible via local WiFi networks. WiderNet copies abundant internet web content to its hard disk and, thus, requires more powerful hardware. From hardware configuration, the internet-offline Solution is somewhat similar to SolarSPELL. However, no detailed information on the software and content configuration of SolarSPELL. The internet-offline solution is somewhat similar to the WiderNet Project incapability to copy and show internet-offline web sites.

In contrast to the WiderNet Project, the internet-offline server includes offline Wikipedia, facilities to do file sharing and streaming video, and much lower cost. The internet-offline solution is a server, not a client. It can show web offline, file sharing, and streaming video. Also, the internet-offline solution includes Kiwix for Wikipedia et al. All at low cost and will be nicely fit into the rural schools' requirements and financial ability.

3. INTERNET-OFFLINE FRAMEWORK

In October 2018, the internet-offline framework was developed and presented for the first time on OnnoCenter Channel in youtube and later presented at FOSSC 2019 in Oman [5]. In general, there are three functions/sections, namely, 1) content controller and accumulator, 2) Operation of internet-offline facilities at school, and 3) students/teachers who are accessing the content using smartphones/computers. The content selection and collection may be centralized and performed by an authoritative body the collected content copies into an SD/MicroSD card and ready to run. Material is then distributed to schools and installed on a local Raspberry Pi server. Teachers and students can access local network facilities such as LAN and WiFi.

All the software used is open source-the detailed note on how to install it freely available on OnnoCenter's wiki in internet-offline topics. To synchronize needed content for offline web content, one uses mainly the commands, 1) rsync a server app to synchronize content [6], and 2) wget a server app to partially copy or copy the entire web content. Scripts for rsync and wget are developed to suit the required needs, such as the URL of the web, and the depth of copied pages. In the internet-offline system, the server has several services. Kiwix is used to display an offline wiki data. Apache, MySQL, PHP provides web services. Samba provides file sharing and video streaming services. The DHCP server is used to give an IP address to LAN clients. BIND-server is used to spoof the hostname and redirect to the local virtual hosting. On the client side, regular tools for accessing the Internet can be used. The users do not have to know that the actual content is on the local server in their school.

4. PRELIMINARY DEVELOPMENT

The author has self-financially started the initial development. The detail documentation is freely released online via OnnoCenter's Wiki and, thus, schools may immediately receive the benefit from the open technology. In general, there are three functions in the internet-offline System, namely, 1) content collector/controller/ accumulator, and content distributor to schools, 2) internet-offline infrastructure at schools, and 3) users mostly students/teachers who access the content via smartphones/computers.

Some of the vital tasks and issues of the content collector/controller/accumulator are as follows 1) determine which content/web needs to retrieved based on the school's curriculum, 2) total content size will be limited to the size of MicroSD. The average affordable MicroSD has around 8-16 Gbyte space, which is a lot for text content, but not much of video content, 3) content should be adjusted to the school's curriculum at different levels/grades, and 4) content should be modified to the teaching methods in Indonesia, there is currently no standard on the teaching methods on using online materials. Not many education faculties provide a practical approach to maximize online teaching material in the learning process. Thus, a teacher's creativity is required to maximize the benefits of existing content.

In-depth content analysis for the internet content selection process needs to be done. Research on an automated content selection process using web crawling techniques with text mining and deep learning classification is currently being performed. One of the most significant advantages of internet-offline Solution is that teachers no longer have to worry about pornographic, inappropriate content, cyberbully, spam, hoax, and unproductive contents from the Internet. Thus, schools may focus on actual teaching only content. Technologically, the process of copying material to make it available offline is relatively simple, and even the Wikipedia community has provided Kiwix for these purposes. To create a web copy of a site into the Raspberry Pi's MicroSD is by using the following commands,

```
mkdir -p /var/www/html/webmirror
cd /var/www/html/webmirror
wget -mkEpnp https://thenameoftheweb.com/
```

It should be performed for all needed webs, after receiving their permission to copy. Next, a client device may need to be spoofed to think it is accessing actual Internet sites, actually the internet-offline redirects the request to its local server. A local BIND Domain Name System must be installed on the internet-offline server and performed the redirection. The zone file of the BIND server has to be configured for each copied web to redirect the Web request to the local internet-offline Server address. Thus, for each web of thenameoftheweb.com, a zone file is created.

The web server Apache configuration needs to be added for each copied web, such as /etc/apache2/ sites-available/thenameoftheweb.com.conf for virtual hosting of the copied web. The configuration parameters are quite simple, such as,

```
ServerName www.thenameoftheweb.com
ServerAlias thenameoftheweb.com
DocumentRoot /usr/local/src/webmirror/thenameoftheweb.com/
```

For each copied web, one needs to write an Apache configuration. Samba may provide file sharing for video, multimedia, docs including for streaming videos. A more detailed configuration of the internet-offline server is obtainable from OnnoCenter's Wiki.

This work uses dnsmasq [7] and hostapd [8] applications to activate the built-in WiFi client in Raspberry Pi into a WiFi access point. The configuration parameter in dnsmasq sets the clients' IP address allocation and the server's IP address. In the hostapd configuration file, several WiFi parameters, such as interface, driver, SSID, WiFi mode, and channel, are set. The school's operator task is only to turn on the system when needed and shuts it down after use. To reduce the MicroSD possible crash, it is advisable to login and performs the proper server shutdown processes.

Students/teachers accessing the content via smartphone/computer may have several notes, such as smartphones, tablets, laptops may be used to locate available teaching materials. For a large number of clients, a school needs to provide additional WiFi access points and LAN cables. A single WiFi access point will effectively serve a maximum of 10 simultaneous devices. To serve simultaneous access, many WiFi access points configured to use 1, 6, or 11 orthogonal channels in 2.4GHz, are needed. The biggest problem is the electrical power to keep smartphones, tablets, and laptops running for an extended period.

5. INTERNET-OFFLINE BENCHMARKING

The internet-offline system is benchmarked to estimate the limitation of the Raspberry Pi3 server. Three different system configurations listed in the Table 1, are evaluated and compared. In this work, we performed several benchmarking processes on the systems by measuring 1) packet per second (PPS) of the network, 2) bandwidth (Mbps), 3) file sharing performance using dbench, 4) operating system performance using UnixBench, 5) Apache Web performance using ab, 6) web Stress test using siege, and 7) power consumption of the Raspberry Pi3 Server. To probe any network bottleneck, the benchmark of the virtual machine will be performed and compared to both via physical LAN connection and direct bridge to host client with no physical network connection.

Table 1. System specification

System	Raspberry Pi3	Asus UN45H MiniPC	VM on VirtualBox
SoC	Broadcom BCM2837		
CPU	4× ARM Cortex-A53, 1.2GHz.	Intel® Braswell Dual-Core Celeron N3000 Processor	Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz. CORE 1-4 adjustable.
GPU	Broadcom VideoCore IV.	Intel® HD Graphics	
RAM	1GB LPDDR2 (900 MHz)	2 GB Dual Channel DDR3L (1600MHz)	512-4096M adjustable.
Networking	10/100 Ethernet, 2.4GHz 802.11n wireless.	100/1000/10/Gigabits Mbps Ethernet, 2.4GHz 802.11 b/g/n wireless.	Virtual Net connection and Physical 100Mbps Ethernet.
OS	Raspbian	Ubuntu Server	Ubuntu Server
Price	US$35-50	US$600-700	US$200-300

5.1. Network throughput

Measurement network throughput (packet per second) is performed by blasting many packets into the network interface of the device under test [22-24]. The simplest option is using the ping command simultaneously running on several instances. However, to get a more accurate result, one may timed blast using a faster hping3 command [22-24]. For a device at 192.168.0.1, the command is as follows,

```
time hping3 192.168.0.1 -q -i u40 --icmp|tail -n10
```

The -i -- interval switch tells the waiting time before the next ping packet. The uX parameter is in a microsecond. The u40 means wait for 40 microseconds before the next ping. In the test, uX number was reduced until packet loss increased by above 1%. To saturate the device under test, the firewall settings are flushed, and the operating system (ICMP) ping rate limit is removed using the command,

```
sysctl net.ipv4.icmp_ratelimit=0
sysctl net.ipv4.icmp_ratemask=0
```

The measured packet per second can be calculated from two (2) times the number of packets received divide by the measured time. The result measured packet per second is as shown in Table 2. The results clearly show that the Raspberry Pi3 LAN network interface is only capable of 25% of the other devices. The Raspberry Pi3 WiFi interface is an order of magnitude lower than its LAN interface, around 3-4% of other devices. In the case of Virtual Machines, different network connection, core, and memory show no significant pps differences.

Table 2. Throughput measurement results

System	-i u	Packet per second (pps)	Note
Raspberry Pi3 LAN	42	30.438	
Raspberry Pi3 WiFi	270	5.740	
Asus UN45H	7	64.371	
VM 1 Core 1000MB	8	136.100	Via LAN
VM 4 Core 4096MB	8	128.715	Via LAN
VM 1 Core 1000MB	9	129.156	Direct bridge no LAN
VM 4 Core 4096MB	9	129.928	Direct bridge no LAN

5.2. Network bandwidth

Network bandwidth measurement is performed by transferring gigabyte zeros through the nc tunnel. On the device under test, a server is run at specific port (e.g., 12345) using nc command,

```
nc -vvlnp 12345 >/dev/null
```

At the client side, connected on the same physical network, we use dd command-line utility for Unix and Unix-like to convert and copy files and pass them out to nc tunnel to the device under test server (e.g., 192.168.0.1) at port 12345 using the following command,

```
dd if=/dev/zero bs=1M count=1K | nc -vvn 192.168.0.1 12345
```

Table 3 shows clearly that all machines with LAN interface, such as the Raspberry Pi3 LAN interface, Asus MiniPC UN45H, and the virtual machines have approximately 100Mbps network bandwidth. The Raspberry Pi3 WiFi interface has a lower approximately 50Mbps network bandwidth. The network bandwidth between the virtual machine and the client on a direct bridge on the same host without any physical LAN interconnection indicates that the actual virtual machine bandwidth is much higher about 2264Mbps and 2376Mbps for single and four core virtual machine, respectively. Thus, the physical interface limits the network bandwidth. It is interesting to note that increasing the core and memory in the virtual machine (VM) does not improve the network bandwidth of the VM.

Table 3. Bandwidth measurement results

System	Mbps	Note
Raspberry Pi3 LAN	93,6	
Raspberry Pi3 WiFi	50,4	
Asus UN45H	94.4	
VM 1 Core 1000MB	93,6	Via LAN
VM 4 Core 4096MB	93,6	Via LAN
VM 1 Core 1000MB	2264	Direct bridge no LAN
VM 4 Core 4096MB	2376	Direct bridge no LAN

5.3. System performance index

Measurement of system performance is done using UnixBench [25, 26]. UnixBench is capable of measuring system performances for multi-core architectures [27]. System benchmarks index score was obtained after measuring the system Execl throughput, file copy with 256, 1024, 4096 buffer size, pipe throughput, pipe-based context switching, process creation, shell scripts with single and eight concurrent processes, system call overhead, Dhrystone 2 using register variables, and double-precision Whetstone. The resulting system benchmarks index score is as shown in Table 4. It shows clearly that the Raspberry Pi and Asus MiniPC UN45H has somewhat similar UnixBench index score. While, the Virtual Machine with one core and four core has twice and four times the UnixBench score as compared to the Raspberry Pi3, respectively.

Table 4. UnixBench index

System	UnixBench Index
RaspberryPi3	415.3
Asus UN45H	592.8
VM 1 Core 1000MB	1035.2
VM 4 Core 4096MB	2576.4

5.4. Apache web server performance

Apache Web Server performance, ApacheBench (ab) is used [28-31]. ApacheBench (ab) is a load testing and benchmark tool for hypertext transfer protocol (HTTP) server. Moreover, it gets installed automatically with Apache web server, or it can be installed separately as Apache utility. It runs off the command line. A quick load testing output obtains in just one minute. ApacheBench (ab) is not stressing the system. Thus, the results would be the same for the different concurrent clients. In this work, 10-150 client's concurrency was measured and find the results are the same for the same system under test. The following Table 5 depict the ApacheBench results.

Table 5. ApacheBench measurement results

	Rpi LAN	Rpi WiFi	Asus	VM 1 Core LAN	VM 4 Core LAN	VM 1 Core no LAN	VM 4 Coreno LAN
Request/Sec(#/sec)	7.58	13.71	172.32	225	226	371	491
Mean Time/Req (ms)	131.96	72.98	5.80	4.43	4.43	2.7	2.0
Transfare Rate (Kb/s)	383.03	691.92	8700.70	11413	11411	18753	24835

It is interesting to note that, although the WiFi interface of the Raspberry Pi3 has lower bandwidth than that of the LAN interface, the RasberryPi3's ApacheBench (ab) results show that WiFi access seems to have a better, double the performance than that of LAN access. Raspberry Pi3 has a capacity of around 7 and 14 requests per second for LAN and WiFi load, respectively. It is an order of magnitude lower than other systems, i.e., Asus, Virtual Machine i5 3GHz single and four-core. Physical interface limits the virtual machine performance. Benchmarking virtual machine via a direct bridge on the same host depict 40-50% higher performance as compared to via physical network interface. There seems to be no performance difference between a single and four cores virtual machine.

5.5. Stressing the web server

To test the web performance limit, Siege is used. Siege is an open source regression test and benchmark utility [32-34]. Siege is configured to read many URLs into memory and simultaneously stress the system. The program reports the total number of hits recorded, bytes transferred, response time, concurrency, and return status. In the test, Siege was run various concurrent clients, to stress various URLs on the wiki web server. Stress tests performed to a Wikipedia mirror in a short 60 seconds.

Figure 1 shows the availability (%) as the system stress using Siege. Above 20 simultaneous clients degrades the Raspberry Pi3' availability. The availability degradation of Asus mini PC began to appear in around 20 clients, and the more severe degradation after more than 80 simultaneous clients. The i5 virtual machine engine seems far more resistant and able to withstand a load of more than 150 clients with only slight availability degradation.

Figure 2 shows the hits received as the number of clients increases. Raspberry Pi3 is only able to accept the lowest hits, around 1000-1500 hits, among the four (4) systems. While Asus Mini PC can accept an order of magnitude hits, around 9,000-10000 hits. The i5 engine can receive above 15000 to 25000 hits, around 25 times the ability of Raspberry Pi3. A four-core system able to receive, about 20%, higher hits as compared to the single core machine.

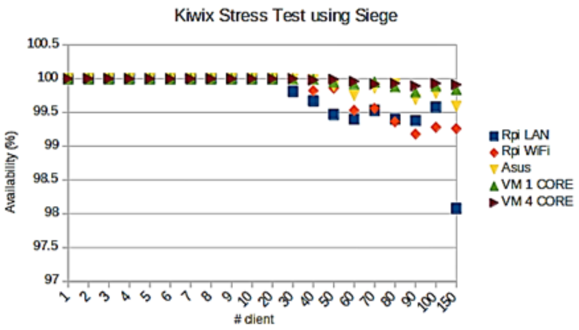

Figure 1. Internet-offline system availability (%)

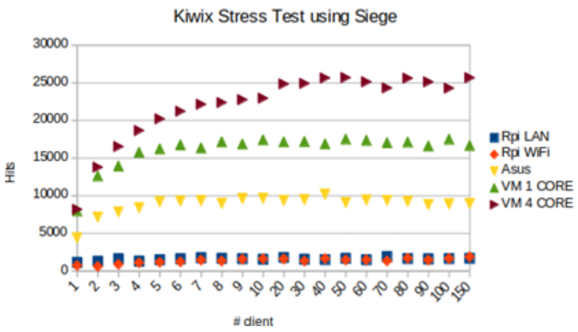

Figure 2. Hits on the internet-offline system

Figure 3 shows successful transactions handled by the system; it is reasonably similar to the characteristics of the hits received by the system. The Raspberry Pi3 is only able to handle around 700-900 transactions. Asus Mini PC can handle well 3000-4000 transactions. The i5 virtual machine can handle well higher transactions, around 6000-8000 transactions. Thus, Raspberry Pi3 is only capable of meeting 10% and 20% capability of i5 and Asus Mini PC, respectively. A four-core system able to receive, about 20%, a higher successful transaction as compared to the single core machine.

Figure 4 shows the transaction rate, in transactions per second, for many clients loading the system. The Raspberry Pi3 is only able to handle around 20-30 transactions per second. Asus Mini PC can handle 5-6 times as compared to Raspberry Pi3 for more than 150 transactions per second. The i5 engine capable of handling much higher loads up to around 250-400 transactions per second or an order of magnitude that of Raspberry Pi3. A four-core system may provide, about 20%, a higher transaction rate as compared to the single-core machine.

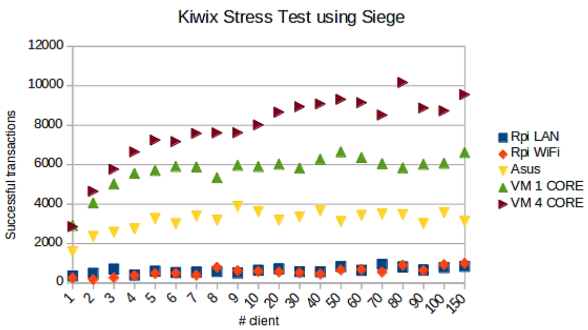

Figure 3. Successful transaction during siege web stress

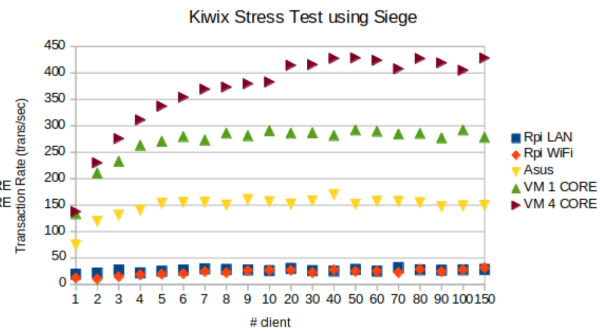

Figure 4. Transaction rate during siege web stress

Figure 5 shows the concurrency capability of the system in handling the clients' load. All systems able to meet the concurrency to be the same as the number of clients up to 20 accessing clients. At 30 clients, concurrency of Raspberry Pi3 and Asus Mini PC is flatten, unable to keep up with the number of accessing clients. At 50 clients, the i5 virtual machine is no longer able to keep up with the number of accessing clients. However, the concurrency of the i5 virtual machine is somewhat degraded but not flat and trying to keep up with the number of accessing client.

Figure 6 shows the failed transaction as a function of the number of clients accessing the system. In the Raspberry Pi and Asus Mini PC, fail transactions began to appear after 30 clients accessing the system. At 40 clients, fail transactions drastically increase. The i5 machine is interestingly able to handle the transactions quite well with fail transactions under five to above 100 clients. Figure 7 displays the longest transaction of the stressed system. All systems seem to handle the load correctly to 20 clients. The Raspberry Pi3 and Asus Mini PC shows a jump in the longest transaction after 30 and more clients load. The i5 virtual machines experience a significant increase in the longest transaction after loading more than 50 clients.

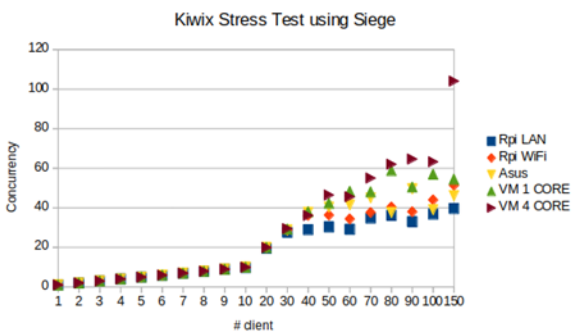

Figure 5. Concurrency during siege web stress

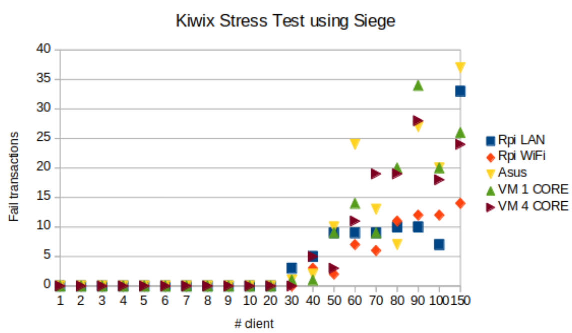

Figure 6. Fail transaction during siege web stress

Figure 8 shows the response time in ms for the incoming transaction as a function of the number of clients. The Raspberry Pi shows a linear increase in response time as a function of numbers of clients, from a single client up to about 20 clients. Above 20 clients, the responds time of Raspberry Pi3 seems to level off. The Asus mini PC and i5 machine seem to be quite resistant to request transactions stress test and can hold respond time under 0.4 seconds up to 100 clients.

It is interesting to note that the network interface limits the performance of the i5 virtual machines. The Table 6 shows the average hits, average transaction rate, average successful transactions, and average fail transactions for single-core and four-core virtual machines with LAN and direct bridging with a client on the same host. It shows that the virtual machine with the direct bridging connection has higher performance and much less fail transaction. The higher performance also observed on the higher core machines.

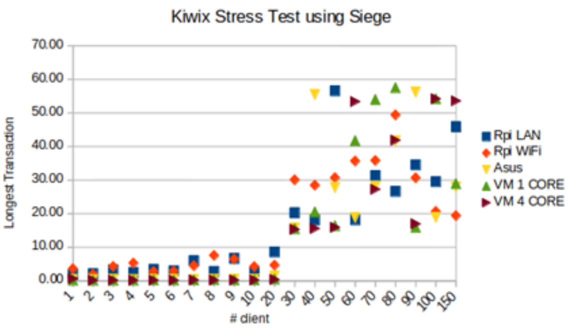

Figure 7. Longest transaction during siege web stress

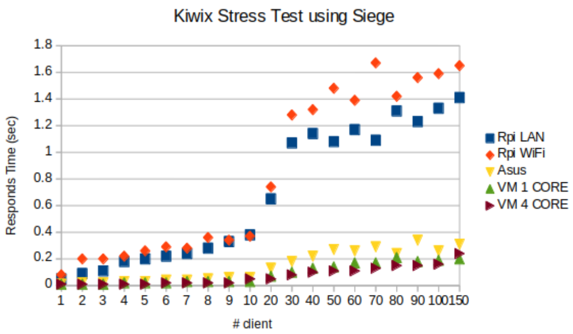

Figure 8. Responds time (sec) during siege web stress

Table 6. Performance comparison LAN and direct bridge during siege web stress

	ave. hits	ave. transaction/sec	ave. successful transactions	ave. fail transactions
1 core LAN	15304	255	5300	14.8
1 core Nolan	17235	287	6084	1.4
4 core LAN	19429	342	6711	14.1
4 core Nolan	22365	373	7844	4.7

5.7. Raspberry Pi3 power consumption

Electrical power is scarce, especially in rural/villages. Thus, one needs to plan better prior using Raspberry Pi3 as a server in rural schools in a limited power environment. Raspberry Pi3 current consumption measured using a small ampere meter installed between the power bank output or the USB power plug and the Raspberry Pi3. The current draw during the web stress test condition and the file-sharing test loads via both LAN as well as WiFi connections is measured. The resulting current profile is as shown in Table 7.

Typical current during smartphone charging is around 1-2 A. Raspberry Pi3 current consumption is found to be lower than the charging current of typical smartphones. When there are no activities, the Raspberry Pi3 consumption is around 0.26 A. Heavy file sharing stress test draw 0.36 A and 0.39 A, for LAN and WiFi stress test, respectively. Massive web access draws 0.43 A and 0.46 A for LAN and WiFi stress test, respectively. WiFi access draws a little bit more current, around 0.03 A, than that of LAN network access.

Most of the affordable power banks sold in online shops have a capacity of around 10000-20000 mAh. With such a large capacity, a fully charged power bank may supply the necessary power for the Raspberry Pi3 server to operate more than six hours. Alternative power resources such as solar cells integrated with the power banks or battery may be of interest for longer operating hours.

Table 7. Raspberry Pi3 power consumption

	Network Stress	WiFi Stress
Standby / no activities	0.26 A	0.26 A
File sharing (dbench)	0.36 A	0.39 A
Web stress (siege)	0.43 A	0.46 A

6. INTERNET-OFFLINE PROPOSED USAGE STRATEGY IN RURAL SCHOOLS

The internet-offline solution based on Raspberry Pi3 has a limited service capacity. A fully charged power may provide 6+ operation hours of a Raspberry Pi3 Server to serve a maximum of 20 clients. The ideal condition is when (1) electrical power is available, (2) the server runs during the school operating hours, and (3) each class has a small WiFi access point that can provide access to students to access the internet-offline server. Since students have any time access to the system, the highest load likely occurs when students have no class activities and do independent learning activities. With the Raspberry Pi3 server, the system will be limited to serve 20 simultaneously access clients. Thus, for a large client based, a larger server is needed. Assembling i5 or i7 machines may provide a low-cost solution for internet-offline servers. If electrical power and funding are limited, then the rural usage strategy for Offline Internet would be:
- Integrate into the computer lab at school.
- Each class participates in the lab class for a particular time. For example, if one lab session about 2 hours, in one day, there will be two (2) maximum class shifts.
- To reduce system load, two students may share one gadget. Therefore, in a single lab class, there will be around 20 gadgets with 40 students.
- Educational content for each class/level should be different. All content in a school may not fit into a single MicroSD. If it does not fit into a single MicroSD, one may split the educational content for each level into separate MicroSDs. Consequently, a primary school may require around six (6) sets of MicroSD. A junior high school requires three (3) sets of MicroSD.

It would of interest to pursue in-depth research work in (1) the content for online study, (3) courses that may provide online enrichment materials to students, and (3) the online assisted teaching techniques.

7. SUMMARY AND CONCLUSION

The main contribution of the paper is in the benchmarking of the low-cost internet-offline solution for rural and village schools. The Rasberry Pi3 server found to be able to serve about 20 clients and able to run for 6+ hours of operation using a low-cost consumer grade power bank. The internet-offline system provides offline information, to overcome the knowledge scarcity in rural and village areas due to lack of Internet access. The required online content copies to a local Raspberry Pi3 Server, and, thus, may be offline viewed by the students. For large capacity, one may use an i5 or i7 PC to keep the price-performance ratio low. The system uses open source, and detailed how-to posted on the OnnoCenter's web and wiki.

It is interesting to note that the network interface limits the performance of virtual machines. The virtual machine with a direct bridging connection has higher performance and much less fail transactions. Also, many core virtual machines demonstrate higher performance results. Web services, such as Kiwix, is limited by the processing power. The physical interface significantly limits the ability to transfer files for clients. For massive streaming and file sharing activities, it would be advisable to binding several network physical interfaces in the server.

The most time-consuming and challenging work is in preparing the teaching materials to be copied to the internet-offline server. From the hardware side, one will face the limitation of MicroSD storage size. It thus limits the scenario and methods in providing the educational material to the students. With minimal space in the MicroSD, one needs to focus on what material needs to be copied to MicroSD so that it is sufficient for teaching processes. One of the most significant advantages of internet-offline Solution for schools is that teachers no longer have to worry about pornographic, inappropriate content, cyberbully, spam, hoax, and much unproductive content from the Internet. Thus, the teachers may focus on teaching the materials. This research is still in the early stage. It hopes to produce a practical solution for schools in rural/village areas. Further work, especially in preparing content in sync with the teaching method, is needed.

REFERENCES

[1] Purbo O. W., "Narrowing the digital divide," *Digital Indonesia: Connectivity and Divergence*, pp. 75-92, 2017.
[2] Malandrino F., Chiasserini C. F., Kirkpatrick S., "Understanding the present and future of cellular networks through crowdsourced traces," *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1-9, 2017.
[3] Kaup F., Fischer F., Hausheer D., "Measuring and predicting cellular network quality on trains," *2017 International Conference on Networked Systems (NetSys)*, pp. 1-8, 2017.
[4] Asrese A. S., Walelgne E. A., Bajpai V., Lutu A., Alay Ö., Ott J., "Measuring Web Quality of Experience in Cellular Networks," *International Conference on Passive and Active Network Measurement*, pp. 18-33, 2019.
[5] Purbo O. W., "Internet-Offline Solution for Rural/Village Schools," *4th Free & Open Source Software Conference (FOSSC'2019-OMAN)*, 2019.
[6] Andrew, et al., "The rsync algorithm," *The Australian National University*, 1996.
[7] Kelley, Simon, "Dnsmasq," *Simon Kelley*, 2008.

[8] M. Jouni, "hostapd: Ieee 802.11 ap, ieee 802.1 x/wpa/wpa2/eap/radius authenticator," 2013. [Online]. Available: https://w1.fi/hostapd/
[9] United States Patent, "United States patent US 6,493,758," 2002.
[10] Shaver R., "Systems and methods for providing a similar offline viewing experience of online web-site content," *United States patent US 8,001,471*, 2011.
[11] Shaver R., "Systems and methods for a single development tool of unified online and offline content providing a similar viewing experience," *United States patent US 8,015,491*, 2011.
[12] Linzy K., Hosman L., "The SolarSPELL Offline Digital library," *IFLAWLIC 2018 Kuala Lumpur*, pp. 1-8, 2018.
[13] Perry A., "Global Education, Global Challenges: Licensing for the New American University," *Against the Grain*, vol. 29, no. 4, 2019.
[14] Oyelude A. A., "What's trending in libraries from the internet cyberspherem alternate technology use," *Library Hi Tech News*, vol. 33, no. 10, pp. 15-6, 2016.
[15] Hosman L., "Impatience as a Virtue: Addressing Persistent ICT-in-Education Challenges in Small Developing Countries," *ICT-Supported Innovations in Small Countries and Developing Regions*, pp. 169-193, 2018.
[16] Maron D., Missen C., Tayag E., Mc Neirney K., "Lo-Fi to Hi-Fi" Crowd Cataloguing: Increasing E-Resource Records and Promoting Metadata Literacy within WiderNet," *iConference 2015 Proceedings*, 2015.
[17] Gafinowitz N. C., "Digital Library Appropriation in the Context of SubSaharan Countries: The Case of eGranary Digital Library Implementation," *Proceedings of the 7th Annual Symposium on Computing for Development*, 2016.
[18] Reed P. J., Kanya R. A., "ICT, Aspirations, and Career Decisions: Does Access to Information Change How Nigerian Students Make Career Decisions?," *iConference 2018 Proceedings*, 2018.
[19] Inefuku H. W., "Globalization, open access, and the democratization of knowledge," *Educause Review*, vol. 52, no. 4, pp. 62-63, 2017.
[20] Lin J., "The sum of all human knowledge in your pocket: Full-text searchable Wikipedia on a Raspberry Pi," *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 85-86, 2015.
[21] Gagneux E., Bowers B., Garcia N., Harkat C., Diouf M., Engome V. P., "Afripédia: rebalancing access and visibility on Wikipedia," *24th BOBCATSSS Conference Proceedings & Abstracts*, 2016.
[22] Luckie M., "Scamper: a scalable and extensible packet prober for active measurement of the internet," *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 239-245, 2010.
[23] Chen B., Pattanaik N., Goulart A., Butler-Purry K. L., Kundur D., "Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed," *2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, pp. 1-6, 2015.
[24] Buchanan B., Flandrin F., Macfarlane R., Graves J., "A methodology to evaluate rate-based intrusion prevention system against distributed denial-of-service (DDoS)," *Cyberforensics*, 2011.
[25] Smith B., Grehan R., Yager T., Niemi D. C., "Byte-unixbench: A Unix benchmark suite," *Technical report*, 2011.
[26] Babu A., Hareesh M. J., Martin J. P., Cherian S., Sastri Y., "System performance evaluation of para virtualization, container virtualization, and full virtualization using xen, openvz, and xenserver," *2014 Fourth International Conference on Advances in Computing and Communications*, pp. 247-250, 2014.
[27] Litayem N., Saoud S. B., "Impact of the Linux real-time enhancements on the system performances for multi-core intel architectures," *International Journal of Computer Applications*, vol. 17, no. 3, pp. 17-23, 2011.
[28] Sheldon M. X., Weissman G. V., "Retrace: Collecting execution trace with virtual machine deterministic replay," *Proceedings of the Third Annual Workshop on Modeling, Benchmarking and Simulation (MoBS 2007)*, 2007.
[29] Wu H., Liu F., Lee R. B., "Cloud Server Benchmarks for Performance Evaluation of New Hardware Architecture," *arXiv preprint arXiv:1603.01352*, pp. 1-5, 2016.
[30] Golasowski M., Slaninová K., Ševčík J., Ptošek V., Vojtek D., "Server-side navigation service benchmarking tool, *Proceedings of the 14th International Symposium on Operational Research*, pp. 85-90, 2017.
[31] Prokofyeva N., Boltunova V., "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems," *Procedia Computer Science*, vol. 104, pp. 51-6, 2017.
[32] Abbas R., Sultan Z., Bhatti S. N., "Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege," *2017 International Conference on Communication Technologies (ComTech)*, pp. 39-44, 2017.
[33] Abbas R., Sultan Z., Bhatti S. N., "Comparative Study of Load Testing Tools: Apache JMeter, HP LoadRunner, Microsoft Visual Studio (TFS), Siege," *International Conference on Computing and Mathematical Sciences ICCMS'2017*, 2017.
[34] Pradeep S., Sharma Y. K., "A Pragmatic Evaluation of Stress and Performance Testing Technologies for Web Based Applications," *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 399-403, 2019.