KTH Royal Institute of Technology

Highly Available and Robust Network Services in Under-served Areas

Jiannan Guo

Master's thesis

Supervisor: Prof. Björn Pehrson, KTH
Robert Olsson, KTH
Dr. Amos Nungu, DIT
Examiner: Prof. Markus Hidell, KTH

Stockholm, 2014

Preface

This thesis project consists of two main parts. The first one is continuity of Serengeti Broadband Network development and second one sets the foundation of establishing robust web servers in rural area. Both projects take place in Serengeti district, northwestern Tanzania.

Overall goal of SBN is to provide robust, affordable broadband local network to provision ICT services such as e-Governing, e-Health and e-Learning. The continuity involves diagnosing and debugging current system, exploring potential risks, and documenting current state for the purpose of administration. Also, it involves development and deployment of new low power router, which stands as next generation of fiber optic routers in SBN. This project is under supervision of KTH, DIT and COSTECH, and main actors are local technicians, government and other public entities such as schools and dispensaries.

The second project aims at populating e-Learning content of Open University of Tanzania (OUT) to remote area. By studying local condition and environment, we are able to identify the challenges that require special attention. Those challenges are then formulated to a set of technical problems, which can be tackled by a variety of available technologies. This part of thesis work stands as the foundation of the project and oulines future development. It is under the supervision of KTH and SPIDER, in cooperation with OUT and DIT.

This report is divided to two parts to address two projects repectively. Each part stands as a standalone and comprehensive report, including all necessary sections to state the problem and the solution.

Contents

List of Figures

List of Tables

Part I

Robust Network Infrastructure in Rural Areas of Tanzania

Chapter 1

Introduction

Information and Communication Technology (ICT) can effectively support poverty alleviation and livelihood enhancement. Bringing ICT in developing counties and under-served areas requires cooperative partnerships and unique research commitments. There exist many outstanding projects and business cases in the field, although not all of them is reproducible, especially when adapted to a different culture and work environment. Two districts have been selected to conduct a pilot project, namely Serengeti Broadband Network, aiming to enhance ICT penetration, build buying power and establish a foundation of rural ICT development. In this thesis report, we cover: 1)A brief introduction and background of project; 2)Current states and challenges; 3)Our approaches and implementation; 4)Future plan. This report stands as a comprehensive documentation of current state of SBN development.

1.1 Background

ICT4RD[1] is designed as a research and business development project, aiming at provisons of ICT services in under-served areas of Tanzania. The project is funded by Swedish International Development Agency (SIDA)[2], and coordinated by Tanzania Commission of Science and Technology (COSTECH)[3]; Dar es Salaam Institute of Technology (DIT)[4] Tanzania; and the Royal Institute of Technology (KTH)[5], Sweden.Two pilot projects are created under ICT4RD, respectively Serengeti Broadband Network development (SBN-development) and Wami project. In this report, we only focus on SBN project.

[1] www.ict4rd.ne.tz
[2] www.sida.se
[3] www.costech.or.tz
[4] www.dit.or.tz
[5] www.kth.se

SBN development aims at building a self-sustained Local Access Network (LAN) convering two districts[**?**]. It hosts services locally, such as VoIP, mails. When upper link exists at any site of the LAN, other parts can also be online. By interconnecting schools, dispensaries and governments, a variety of applications are proposed and implemented over the network, such as e-learning, e-governing and e-health.

It is the seventh year of SBN project. During these years, enormous research effort has been contributed from partners to establish the broadband island. Currently, SBN consists of two main components. To the east side, optical fiber is used to connect three towns in two districts, namely Bunda, Nata and Mugumu respectively. And to the west, wireless technology is used to exten the network for another 100km.

1.2 Related Work

There are many successful rural ICT projects which stand as good references. Several of them are presented here, mainly focusing on their technical landscapes.

- **Macha Project** Macha is located in the Southern province of Zambia, with a population of 135,000 (c.2007). In Macha project[1], VSAT is introduced to source Internet connection from satelite. PCs are connected via local wireless network. When connecting remote area with no infrastructure available, this set of technologies is most intuitive and economical.

- **Nepal Wireless** In Nepal Wireless project, the main goal is to provide Internet connection for schools. The source of Internet comes from nearest town and then distributed via wireless links. Because of the unique location and landform, harsh natural environment becomes one of the greatest challenges, which directly affects power supply[2].

Inspiring projects are not limited to ones mentioned above, there also exist innovative approaches to bring connection to developing areas, such as physical transport in DakNet[3].

1.3 Problem Statement

ICT development in rural areas is a challenging task which requires innovations on both technical and managerial sides. We identify following main obstacles that limit technology deployment:

- Low affordability. In rural development, expenditure has alwasy been a critical factor, not only during the phase of procurement, but also

maintainance and refurbishment. Necessary trade-off needs to be made between capacity and cost.

- Poor supply chain, especially power shortage. The lack of electricity is always a limitation in rural development. In Tanzaina, only 14% of the country is electrified and the figure reduces to 2% in the case of rural area[**?**]. Innovative power source such as solar is highly desired. Even those sites along the power line are also challenged by frequent power outage and voltage spikes.
- Harsh environment. In rural sites, temperature is considerably higher than recommanded operational level, especially at those sites where equipments boxes are mounted outdoor.

1.4 Approach

Special requirements need to be treated differently with innovative approaches. In SBN development project, we apply iterative approaches to test new solutions. Different components are gradually replaced with newly developed technologies. we carefully select off-the-shell hardware and open source software to reduce the cost and enhance rebustness. To attack the problem of power supply, we run our equipments over heterogeneous power source, including sink device such as battery or super capacitor. By reducing the power consumption, we minimize discharge cycles and prolong battery life.

1.5 outline

Remaining content is organized as following: Chapter 2 starts with an overview of previous projects of SBN. It then explains current network topology and administration; Chapter 3 demostrates the design of low power router. Deployment and testing are discussed in this chapter as well. In chapter 4, we draw our conclusion and propose improvements that could be done in the future.

Chapter 2

An Overview of SBN

In rural ICT development, attentions should be focused on affordability while providing reasonable capacity. This requires pioneers to carefully investigate local condition and foresee potential risks and opportunities. Collaborative approaches sometimes result in extraordinary outcomes. In this chapter, we aim at providing a comprehensive vision of ICT development in rural area, including available technologies, key challenges and local conditions. We then present the current state of SBN development.

2.1 Communication Technologies in Rural Development

Technology selection in rural ICT development is highly affected by physical environment and the demand of services. Conditions are often different from one site to another, hence not replicable in its entirety. As introduced in section **??**, Macha project represents a typical setting of rural ICT envrionment, while Nipel project serves as a good example of distribution of network. In this section, Four communication technologies are presented and evaluated against SBN environment.

Optical fiber links are favorable due to its capacity and durability. Although deployment and maintainence require special tools and skills, and civil work involved is immense. Building a fiber line in rural area demands innovative cooperation to distibute risk and cost, as well as sharing the benefit. In the case of SBN development, during the time that Tanzania set up the power line between two districts, a 140km optical fiber link was also established along the power transmission line and owned by Tanzanian power company, TANESCO[1]. The fiber was donated to ICT4RD in exchange for network connection. To distribute fiber backbone network, Low power routers that support both optical fiber and copper links are developed. More

[1] http://www.tanesco.co.tz/

details can be found in section **??**

Terrestrial Wireless is an optimal approach in rural first-mile delivery comparing to tranditional landline communication. It is easier to deploy and highly customizable to adapt to different landform. Among various wireless technologies, IEEE 802.11 family (more commonly known as WiFi) running in license-free 2.4GHz or 5GHz offers satisfactory bandwidth while eliminating the cost of frequency registration. In SBN, WiFi is intensively deployed at first mile to distribute connection from optical fiber link to end users. Furthermore, optical fiber backbone is extended by WiBACK[2] network which enable decent video streaming at remote sites.

Many rural ICT projects deploy **Very Small Aperture Terminals** to source the Internet from satellite. This approach is highly favorable for remote sites where establishment of infrastructure is simply not feasible. Although most of satellite link come at a very high price and limited bandwidth. In SBN, TTCL extends their data service to Bunda town, where we link our LAN to the Internet.

There exist novel and untraditional approaches in rural ICT development to link remote villiges, such as **Delay Tolerent Network** in DakNet[3], which may physically transport bulk of data by vehicles. In spite of high bandwidth, latency introduced is not suitable for real-time applications such as video call.

2.2 Key Challenges

2.2 Challenges in rural ICT development are different from that in urban areas and entail innovations in many ways. Some common obstacles such as poor supply chain, weak buying power and unsatisfactory knowledge base are also encountered by other projects. Although SBN comes across some unique requirements. We are able to identify following key challenges from the observation: *Poor supply chain, especially power supply* Overall electrification rate in Tanzania is 14% and less than 3% in rural areas. Even at those sites where electricity is accessible, power outage is frequent and sometime harmful due to voltage surge. To power up electronic devices such as routers and transceivers, PV system is intensively deployed to source power from solar and deposit it into batteries that can be used over night.

1. Weak knowledge base and lack of necessary skills
   The lack of capable human resources to operate and maintain the network greatly limits the development. Although we find leadership and management skills more critical in rural development. The project is significantly blocked by the facts that managerial entities are slow in reaction and reluctant to collaborate. More of this topic can be found

[2] www.wiback.org

in [4] although the discussion of project organization and management is beyond the scope of this report.

2. Harsh ambient environment
   Due to limited budget and confined spaces, equipment boxes are often mounted on the pole along with the radio or fiber line. Electronic devices and batteries are placed inside the box where the temperature is always unfavourably high during daytime. This is also the time when battery can be charged by solar power. Study shows that each 8℃rise in temperature cuts the life of a sealed lead acid battery in half[3], whilst a temperature of 25℃ is recommended.

3. Poor buying power
   As the overall objective is to digitalize under-served land, the fact of affordability problems is presupposed. Low cost off-the-shelf hardwares and open source software are chosen to cope with the limited budget. To gain a better price-performance ratio, we conduct a benchmark of several platforms in section 3.2.

4. Sparse population density
   In SBN, population to be covered spread over a vast area. This entails significant effort to backhaul network from those remote sites.

2.3 SBN Topology

2.3.1 Fiber Optic Line

SBN consists of three networks locating in Bunda, Nata and Mugumu respectively. They are interconnected by fiber optic line. An overview of SBN topology is shown in Figure 2.1. The router currently in use is Supermicro

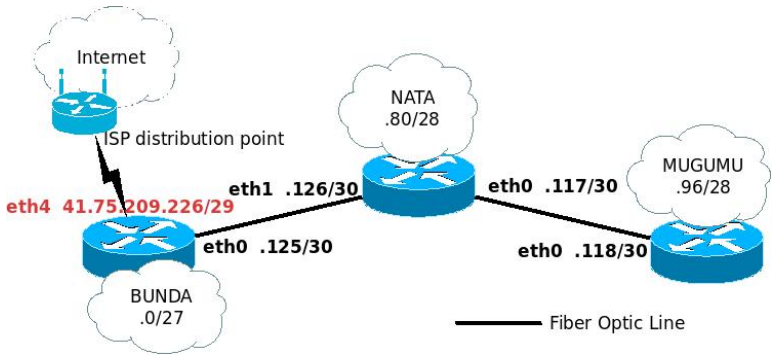


Figure 2.1: An Overview of SBN

X7SPA based system with integrated power supply unit. More of system

[3]http://batteryuniversity.com

design is discussed in Section 3.2. The network is then distributed to end users through terrestrial wireless devices.

2.3.2 WiBACK Network

WiBACK is used to extend Bunda network to the west. WiBACK is essencially relay agent based on MPLS traffic forwarding. It is easy and fast to deploy and features QoS-provisioning, auto-configuration, self-management and self-healing. In SBN context, WiBACK network relies on 802.11 standards and operates in 5GHz domain.

The initial plan was to achieve a topology shown in Figure 2.2.

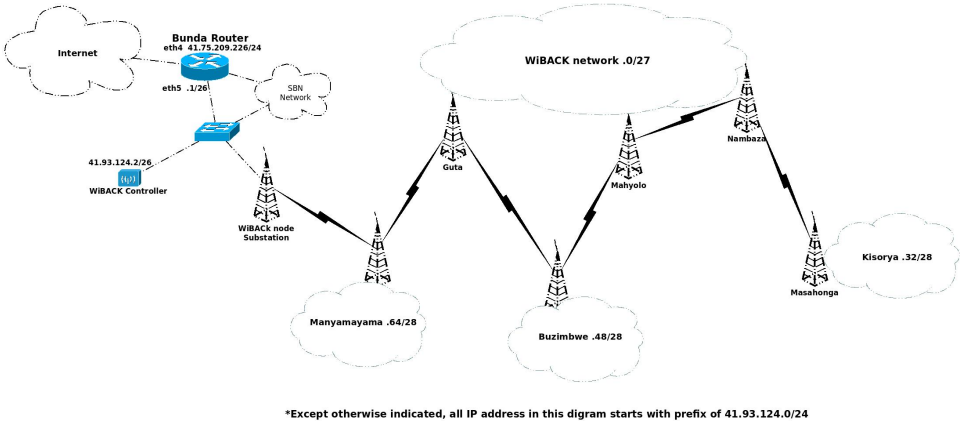


Figure 2.2: The initial plan of WiBACK network

2.3.3 IP Allocation

IP allocation is listed in Table 2.1. A visualization using Hilbert Curve can be found in Appendix X.

Domain	IP	Remarks
Administration	0/29	Administration
WiBACK	0/27	Reserved for other WiBACK network
	32/28	Kisorya (AIRC) network
	48/28	Buzimbwe (Kibara network)
	64/28	Manyamayama
Serengeti Fiber	80/28	Nata
Network	96/28	Mugumu
	80/28	Nata
	116/30	Nata – Mugumu router
	124/30	Bunda – Nata router
Odroid Testing	144/28	Odroid Network Testing (Internal Links)
Tunnels	128/28	Tunnels

Table 2.1: IP allocation of SBN

Chapter 3

Toward Low Power

In this chapter, we introduce a recently developed low-power router. We start by illustrating the motivation of advancing to low power system. Then we present the design and how it is tested in SBN context.

3.1 Motivation

As pointed out in section 2.2, photovoltaic system is intensively adopted in SBN due to poor or non-existing power supply. The choice of battery is critical in designing an effecient PV system. Within the context of SBN, most batteries available at local market are lead-acid batteries intended for cars and motors, more commonly known as "starting battery". They are designed to supply a large current for a short time to start engine, instead of being deeply discharged with mild current.

In a photovoltaic system, the battery can only be charged during daytime, and may be drained generously with heavy load over night. Deep discharges shorten cycle life of these batteries due to sulfation and crystalization. Elevated internal resistance also makes charging harder. Hence, deep cycle batteries are preferred in PV systems. Unfortunately they are not widely available in Tanzania and come with a relatively higher price.

To prolong the life of starting batteries in PV systems, deep discharges should be avoided by either expanding battery capacity (using a larger battery) or reducing load (drawing less energy from batteries). Given capacity of battery $C(Ah)$, output voltage of battery V, power consumption $P(watt)$ and discharging time t, Depth of Discharge (DoD) can be denoted as:

$$DoD = \frac{Pt}{CV} \tag{3.1}$$

Table **??** shows the power consumption of several typical equipments in SBN. Table **??** reveals the approximate depth of discharge (DoD) of 100Ah and 50Ah batteries for 16 hours[1] under various power consumptions.

Device	Power Consumption
Cisco 2950 series switch	30W
Supermicro X7SPA Montherboard	14W
Ubiquiti Nanostation	8W
Odroid U2	5W

Table 3.1: Power consumptions of frequently used equipments in SBN

	50Ah	100Ah
40W	Over-discharged	53.3%
30W	80%	40 %
20W	53%	26.7%
10W	26.7%	13.3%

Table 3.2: Depth of Discharge for different batteries under different load

Adding batteries leads to more investment and more labor work for mantenance. Thus, we strive to the lower power consumption (*watt*) while meeting the demand of capacity (*bps*).

3.2 System Design

In the initial phase of SBN, connection on fiber-optic line is enabled by Cisco 2950 series switch with fiber-optic SFP ports[2]. An inverter with multiple power input is used to connect DC from power grid, battery and switch. The drawbacks of this design are clear:

- The whole system is energy-hungry, given high power consumption of switch and low efficiency of inverter.
- There is a lack of cut-off voltage protection for batteries.
- Significant room is consumed to store these equipments.

Our low-power optical fibre router design includes a motherboard, network interface card(s)(NIC) which supports one or more optical SFP ports and integrated charge controller. The montheboard of first generation router includes following components:

- Motherboard: Intel/ATOM-based Supermicro X7SA, supporting PCIe I/O-bus.

[1]A PV system in Serengeti area can effectively accept solar power for 7 8 hours a day, which leaves approximately 16 hours with battery as the only power source.

[2]http://www.cisco.com/c/en/us/products/switches/catalyst-2950-series-switches/index.html

- NIC: Interface master Niagara 82048, providing four GIGA SFP ports with digital optical monitoring.

The idling power consumption of the Motherboard is 14W and for the NIC 8W. The differences between idling and full-load is negligible. A bandwidth of 1Gbps is fulfilled in this design, although the power consumption is still not optimal according to Table **??**, especially when distribution radio is taken into consideration. We refer this version as Supermicro router.

To further minimize power consumption while maintaining satisfactory bandwidth, a new version of low-power router is developed with following components:

- Motherboard: Odroid U3[3] with Arm-based Exynos4412 Quad-core processor.
- NIC: one 10/100Mbps Ethernet with RJ45 pack. Two fibergecko 100 card[4] with 100/1000Mbps SFP port and USB2.0 on I/O-bus.

We refer this version as Odroid-router. A comparison of Supermicro router and Odroid router is shown in Figure 3.1.

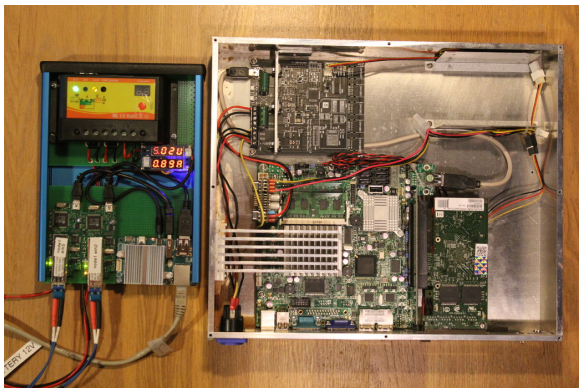


Figure 3.1: The first generation low-power router to the right and the second to the left

The power consumption of new design is approximately 5W and the difference between idling and full-load is negligible. The power consumption is significantly reduced with a compromise of bandwdith. The downgrade of speed is actually forced by the bottleneck of USB2.0 port as the I/O-bus. Local loop-back in the Exynos4412 quad-core processor reveals a communication capacity of 5Gbps. On the other hand, USB3.0 adds a new transfer mode "SuperSpeed" which is capable of tranferring data at up to 5Gbps.

[3]http://www.hardkernel.com/main/products/prdt_info.php?g_code=g138745696275
[4]http://www.lyconsys.com/download/datasheet_fibergecko100_eng.pdf

Hence, the possibility of upgrading the service back to 1Gbps or higher is promised, given capable motherboard and NIC cards.

The choice of motherboard is based on a benchmark of several candidates listed in Table 3.3. The benchmark is conducted using LMBench toolset[5] and the result is revealed in Figure 7. Packet forwarding is essentially based on table lookup, whose performance is impacted by memory latency and bandwidth. The lower latency and the higher bandwidth, the better. They are presented perspectively in the top and middle diagram of Figure 7.

	CPU Freq	Size ($mm \times mm$)	Weight
Raspberry Pi B	0.7GHz	85.6×56.5	45g
Supermicro X7SPA	1.8GHz	170.2×170.2	
Odroid U3	1.7GHz	83.0×48.0	48g
Alix 2d3	0.5GHz	152.4×152.4	
BeagleBone Black	1.0GHz	86.4×53.3	40g

Table 3.3: A list of router motherboard candidates

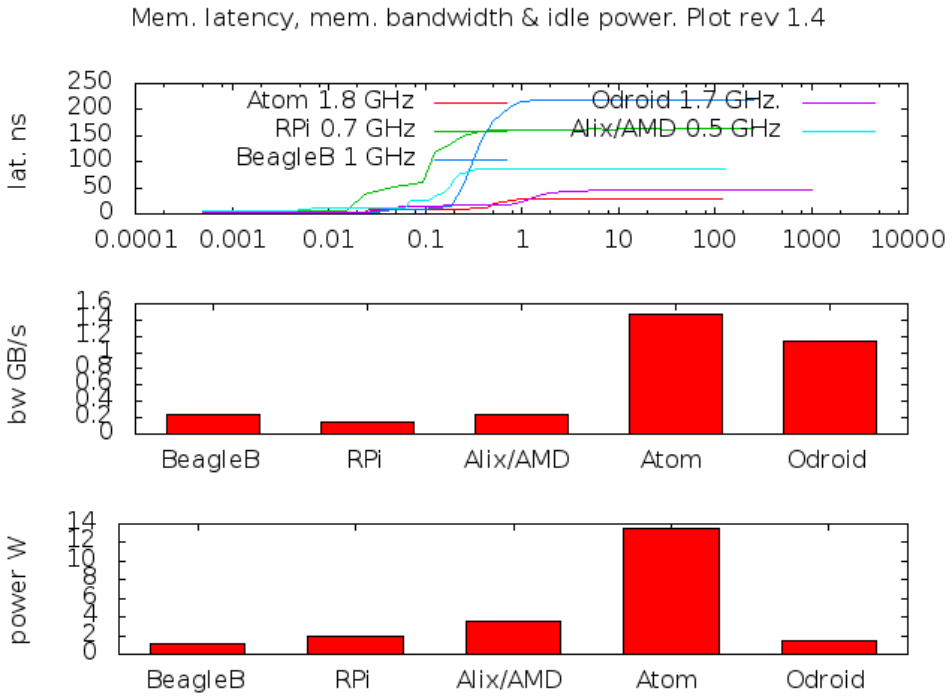


Figure 3.2: A benckmark of five low power platforms (top: memory latency; middle: memory bandwidth; bottom: idling power consumption

[5]http://www.bitmover.com/lmbench/

Intel/ATOM-based system has the lowest memory latency although the data of Odroid comes very close to it. As for memory bandwidth, ATOM system is still the best, which is followed closely by Odroid.

The idle power consumption in the bottom plot is measured without NICs. The Atom system consumes 14W, the Alix board 3.5W and other three platforms consume 1.4W, about 10% of Atom system. When adding two NICs to Odroid, the total power consumption becomes 4.57W when idling and 5.25W when fully loaded by forwarding at speed of 95Mbps.

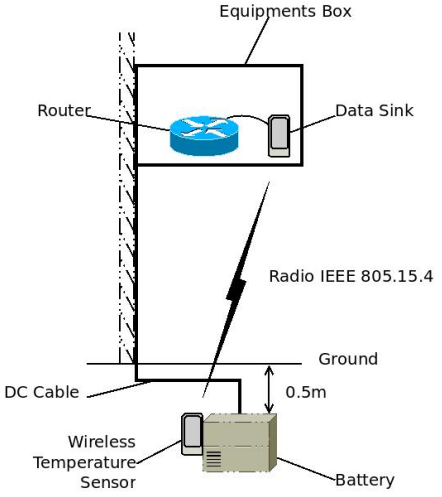
The result becomes clear by comparing five platforms that Atom system has the best performance in terms of memory latency and bandwidth, although followed closely by Odroid with significantly reduced power consumption. Odroid is also smaller in size and weight.

The operating system used previously in Supermicro router is Bifrost Linux distribution[6]. A customized Debian-based distribution with added network performance test utilities is used in Odroid router.

3.3 Deployment and Testing

3.3.1 Temperature Profiling at Nata site

To further investigate the overheating problem addressed in Section 2.2 and verify the proposal of burying batteries underground, wireless sensors[7] are deployed according to the diagram shown in Figure 3.3



(a) Setup to profile ambient temperature

(b) Burying battery at Nata site

Figure 3.3: Temperature Profiling at Serengeti Nata site

[6]http://bifrost.slu.se/
[7]http://herjulf.se/products/WSN/sensors/

A wireless sensor node is buried along with a 100Ah battery. The sensor draws negligible current directly from battery and periodically report temperature data to sink node via IEEE 802.15.4. The sink node then send data to listening daemon running in low power router, in which data is accumulated for further analysis. With collected data, we plot temperature trend of 6 days at Nata site, see Figure 3.4.

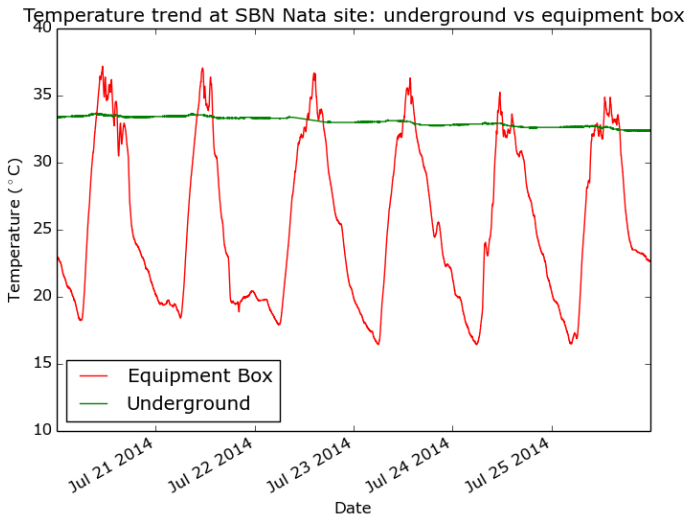


Figure 3.4: Temperature trend at Nata

The temperature in the equipment box is consistent with our observation. In addition, average temperatures are considerably uniform thourghout the whole year in Serengeti area. Hence, battery life can be prolonged if stored elsewhere cooler. However, temperatures under earth contradict with our expectation. Rather than stabling at 25℃[5], temperature at 0.5m below surface stays at a constant level of 33℃.

Hence, other approaches to bring down the battery temperature are yet to be further explored.

3.3.2 Network Topology and Setup

To test Odroid routers, three of them are deployed at Bunda, Nata and Mugumu sites respectively, along with Supermicro routers that are in service. The new topology after adding those routers is shown in Figure 3.5. A subnet of 41.93.124.144/28 is allocated for testing.

As introduced in Section 2.3, SBN fiber line is comprised of two segments: Bunda-to-Nata and Nata-to-Mugumu, respectively. In previous setting, two segments are linked by the Supermicro router at Nata. As a result, this

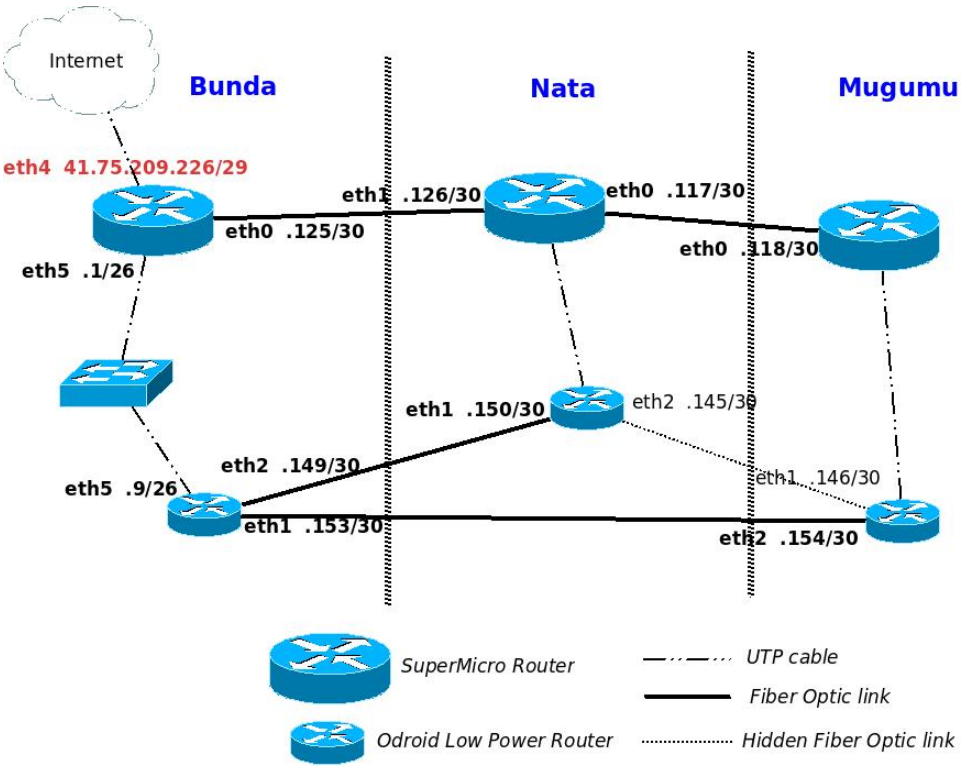


Figure 3.5: Topology with odroid routers

router becomes the single-point-of-failure (SPOF), whose failure causes disconnections of both Nata and Mugumu network. To eliminate the SPOF, it is proposed to have a fiber trand bypassing Nata, which links Bunda and Mugumu directly. Following restrictions need to be considered:

- The optical fiber line is poorly installed and only few of them are usable.
- Even though necessary information is partially documented[8]. For single mode fiber, power loss of 0.5dBm/km for 1310nm is expected and 0.4dBm/km for 1550nm. The distances of two segments are 89km and 45km, which indicate $\tilde{4}$5dBm power loss between Bunda and Nata, and $\tilde{2}$2dBm loss between Nata and Mugumu. According to the documentation, fiber loss is less than that prediction, although abrupt discontinuities result in much power loss and noise observed at the end. In addition, power budget of SFP transceivers on Supermicro router is limited and could not tolerate the power loss combined. Thus, it is

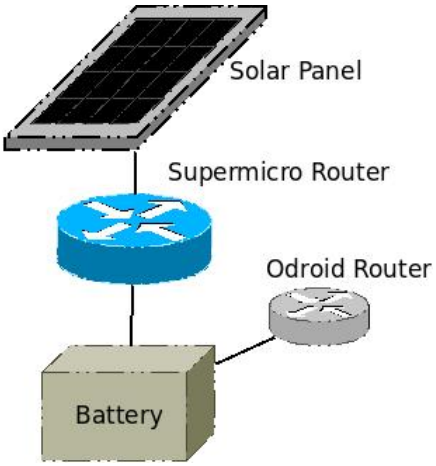
[8]RESEARCH AND DEVELOPMENT PROJECT ON INFORMATION AND COMMUNICATION TECHNOLOGY FOR RURAL DEVELOPMENT (ICT4RD) IN TANZANIA - Optical Fiber Commissioning Report Serengeti Site

hard to bypass Nata with Supermicro routers.

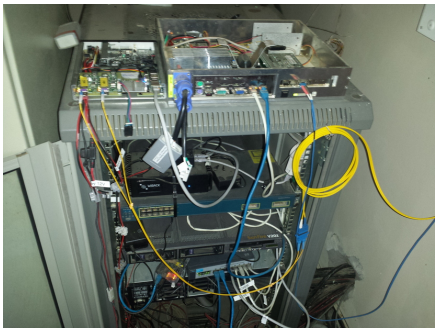
- Without proper tools, locating those fiber trands and testing are non-trivial and time-consuming tasks, especially at remote sites where the fiber spans over one hundred kilometers.

We use duplex SFP transceivers with higher power budget to reduce the number of fiber trands needed and to compensate for power loss. We are able to achieve a toplogy shown in Figure 3.5. How fibers are wired is documented in Appendix X. To be noted, all the IP addresses in this diagram are within the subnet of 41.93.124.0/24, unless labelled otherwise. For example, eth5 .1/26 indicates an IP address of 41.93.124.1/26 on NIC eth5. Routes are statically configured, for routing program is currently not enabled.

Odroid router comes with complete power supply circuit, as well as Supermicro router. Figure 3.6 depicts how solar panel and battery are wired. Low Voltage Disconnect (LVD) level of Odroid router charge controller is set to be slightly lower than that of Supermicro router, and results in later shutdown of Odroid router when battery being exhausted.



(a) Supermicro and Odroid routers share the power supply system



(b) Equipments at Bunda substation

Figure 3.6: Cable solution of Supermicro and Odroid routers to share power supply system

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In current phase of SBN development, objectives mainly consist of following activities: 1)diagnose and refurbish individual components of the network; 2)investigate power supply system to enhance robustness; 3)design and deploy low power routers; 4)log and document these activities for future reference. We are able to identify existing problems within the system and to achieve a comprehensive understanding of local condition.

4.2 Future Work

4.2.1 Routability

Public IP addresses of a /24 network are allocated to SBN. Although, as a broadband island, this network is behind NAT while exposing few IPs assigned by ISP, as depicted in Figure 4.1. However, global routability is

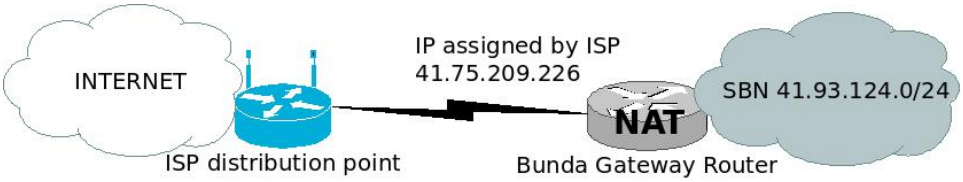


Figure 4.1: SBN /24 network behind NAT

desired to for necessary service being accessible from outside of SBN domain. To achieve this, three approaches are proposed.

- GRE Tunnel
  IP addresses of 41.93.124.0/24 are hold by African Network Information Center (AFRINIC) and managed by TERNET. Routability can be achieved by linking broadband island to another TERNET entity

which is capable of advertising the route. A complete plan can be found in Figure 4.2. Traffic for 41.93.124.0/24 is routed to 41.93.124.1 via GRE tunnel between Bunda gateway router and TERNET gateway router.

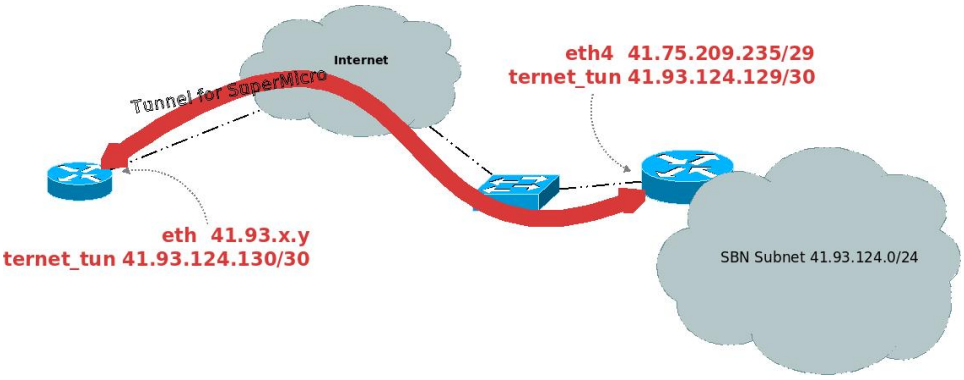


Figure 4.2: GRE tunnel between TERNET gateway and SBN Bunda gateway

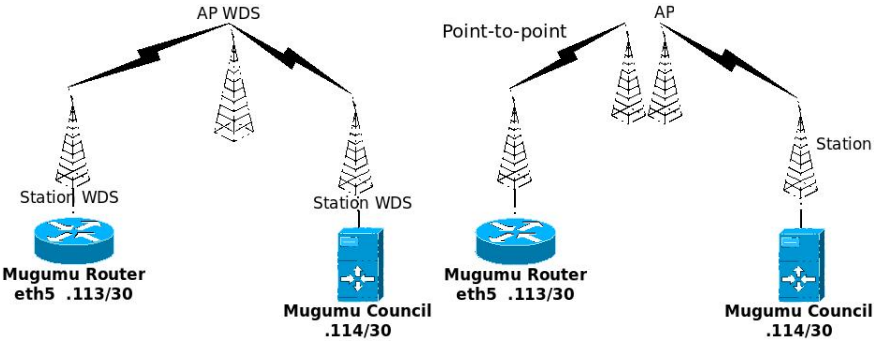
- Local ISP
  Let the ISP be reponsible for the provision of routability, more specifically, advertising 41.93.124.0/24 network at their BGP router. This approach is the most intuitive and natural one, although it requires negotiation with ISP and might cause harmful traffic storm if misconfigured. On the other hand, two ISPs are available and there exists possibilities of switching back and forth. Therefore, this solution is not as flexible as previous one.

- Port Forwarding
  Port forwarding is the most lightweight and cheapest solution, which is currently being used in SBN. By adding proper iptables rules to Bunda gateway router, internal services could be accessed by requesting gateway address on specific port. However it should only be considered as a temporary solution, for it is hard to manage and not scalable.

4.2.2 Wireless distribution restructure

while distributing the network from fiber line to end-users, wireless signal is relayed by an antenna on geographic high point to obtain better coverage. A typical setting is shown in Figure 4.3a. Three radioes are all configured to operate in Wireless Distribution System (WDS) mode[1], however the bandwidth in this scenario is halved when two wireless hops exist in between.

[1]http://wiki.openwrt.org/doc/howto/clientmode



(a) Wireless Distribution System at Mugumu

(b) Distribution of network with point-to-point link at Mugumu

Figure 4.3: Wireless Distribution System

Hence, it is proposed to set up a separate point-to-point link between fiber station and access point, which result in a topology in Figure 4.3b. The point-to-point link is preferably operating in 5GHz domain to minimize inteference.

4.2.3 Power Circuit Failure

We observe occasional abnormal boot while testing Odroid router. When the system is recovered from power failure, Odroid platform sometimes, however, refuses to boot even though blue led (power light) is on, unless power pin is plugged out and back. The issue is yet to be investigated in laboratory.

4.2.4 Network Monitoring and Administration

To manage the system more efficiently, monitoring tools are yet to be deployed in the routers and other devices. There exist several Network Operations Centers in DIT, TERNET and KTH. Zabbix is deployed at some parts of the system although the installation is still incomplete. The task of configuring monitoring system propoerly is left to future work.

Part II

Highly Available Web Services

Chapter 5

Introduction

In this chapter, we introduce the motivation to build a highly available web service in rural area. We illustrate the result of observation and identify the key problems.

In chapter 6, we start by formulating the problem into a multi-master system, and then investigate several available technologies that address this challenge. We examine them against our unique requirement and select one as our basis. We then propose and explain our approach on top of it.

In chapter 7, we illustrate the idea of using ARM-based hardware to achieve low-power consumption web service. To study the capacity of hardware, we benchmark the performance of each components of web service on different platform. According to the result, we propose a cluster than can be easily scaled out to serve more users.

In chapter 8, we draw our conclusion and shed light on future work.

5.1 Background

Affordable, yet stable web services are highly desired in rural area, as a mean to alleviate digital divide and improve life quality. When it comes to under-developed regions in Africa, requirements and conditions need to be carefully assessed and analyzed, for that challenges could be unique and dramatically different than metropolitan.

5.2 E-learning for Open University of Tanzania

Open University of Tanzania (OUT) [6] is the first university of East Africa Region to provide open and distance learning programmes. To distribute course content through the whole country, Moodle has been chosen as underlying digital resource management platform. Moodle[7] is an open-source industrial-level online learning platform and resource management system. As a typical data-driven web service, Moodle runs over an underlying database

and assemble its webpages on-the-fly based on user requests. It is written in PHP and heavily tested against Apache, Nginx and MySQL. At present, the platform is running as a standalone web service in a central server and mainly serve static content such as PDF, Text and Slides. Although OUT has the vision to introduce multi-media materials to enhance education quality. OUT also establishes learning centers in major cities and towns all over the whole country and is ambitious to extend to a larger scale. An emerging obstacle is to provide services in remote areas with poor network connection and bandwidth.

5.3 Problem Identification

As part of the project, we investigated local conditions and needs within the scale of Serengeti Broadband Network, especially in areas with evident demands of services and lack of infrastructures. And we were able to identify following challenges:

5.3.1 Power Outages

Power grid in rural areas of Tanzania is so unreliable that UPS for critical device is almost a must. While people are gradually adapting to mobile platforms, such as smart phones and tablets, backbone infrastructures are also required to be more persistent. Equipments powered up by solar and battery are highly desired due to cost-efficiency. Although power consumption need to be optimized in this circumstance in order to prolong battery life and improve reliability.

5.3.2 Poor network quality and frequent failure

Although local network is operated by ICT4RD project and can be fairly reliable, uplink is still depending on national-wide ISP and is somewhat unpredictable according to our observation. Network failure could occur anytime and can last for random period (several minutes to several days). Those web services that depend on a central server are apparently not accessible during the failure. On the other hand, the uplink can be very narrow due to poor infrastructure and limited budget. It could be difficult to squeeze multimedia services into such bandwidth.

To better illustrate this problem, suppose a typical setting in Figure 5.1. Major backbone components in this LAN are interconnected through fiber-optical lines, and network is distributed to users through WiFi or Ethernet. The LAN is linked to the Internet through ISP distribution link and central server resides on the otherside of the Internet.

Due to limited budget and ISP capacity, the upper link is equipped with an average bandwidth of 2 4Mbps which is shared among all users in

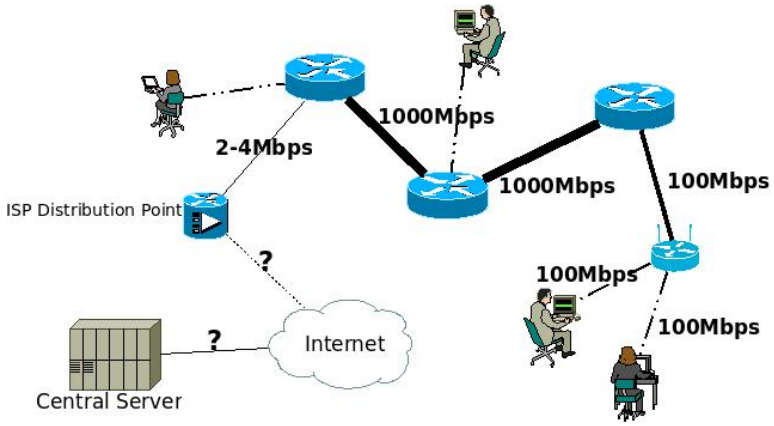


Figure 5.1: A Typical Setting of Rural Local Access Network (LAN)

LAN. While a minimum bandwidth of 1.5Mbps is recommended for video streaming, it is difficult for users to get decent service from central server. To worsen the situation, the upper link is somewhat unpreditable, which leads to the isolation of LAN, as shown in 5.2.

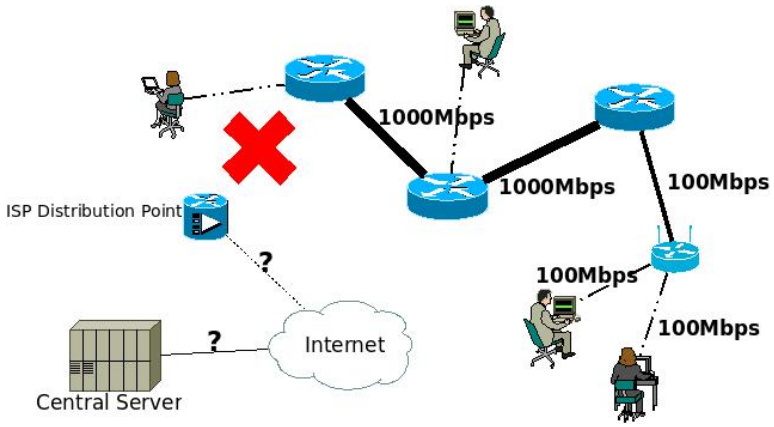


Figure 5.2: Uplink Failure leads to the isolation of LAN

On the other hand, components in the LAN can also break down which leads to network separation, see Figure 5.3. In the first case, multi-media content can hardly reach end users. And in other two cases, users cannot get service at all.

5.3.3 Limited budget

Cost is an essential factor during rural ICT development. Given relatively smaller user base and weaker demand, equipments need to be chosen wisely. Although future maintainence and development also need to be considered.

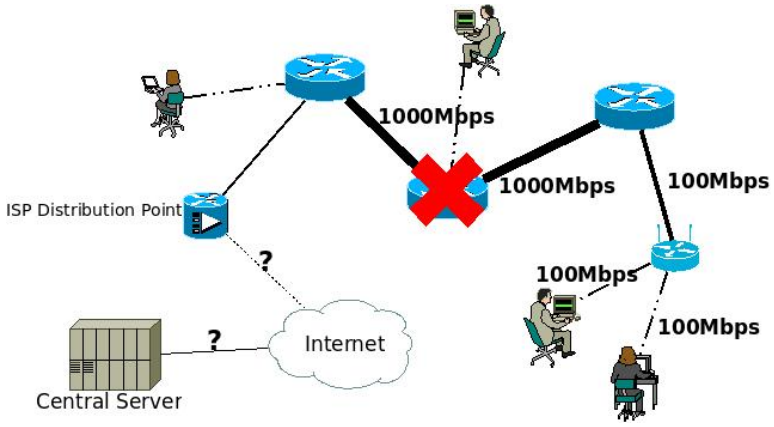


Figure 5.3: Network Separation due to Component Failure

Chapter 6

Adapt to Frequent Network Failure and Limited Bandwidth

In this chapter, problems are further decomposed and analyzed. To prevent reinventing-the-wheel, a variety of possible solutions are proposed and investigated, whereas focus has been put into our unique requirements.

6.1 A closer look at the problem

As introduced in section 5.2, Moodle is deployed as underlying course management system for OUT E-learning platform. Moodle is an open sourse project written in PHP and well-documented[8][9]. Similiar to other web applications, it can be deployed in a typical LAMP or LNMP stack. In this chapter, we mainly focus on possible solutions for two problems stated previously, and leave the choice of actual server to chapter 7

Moodle is a typical database-driven web application where all the pages are generated on-the-fly based on user request. The whole application is composed of three main components:

- PHP source code, typically in `/var/www/moodle/`
- A database to store data or metadata including site configuration, student information, course details, events, etc. There exist volatile tables in Moodle database which store sessions and temparory information.
- A directory to store materials and resources, as well as cache and temparory files. Typically it is named as `moodledata/`

The problem addressed previously can be simplified and modelised as following, see Figure 6.1. Each node in the model denotes a local server/proxy and has a certain amount of users associated with it.

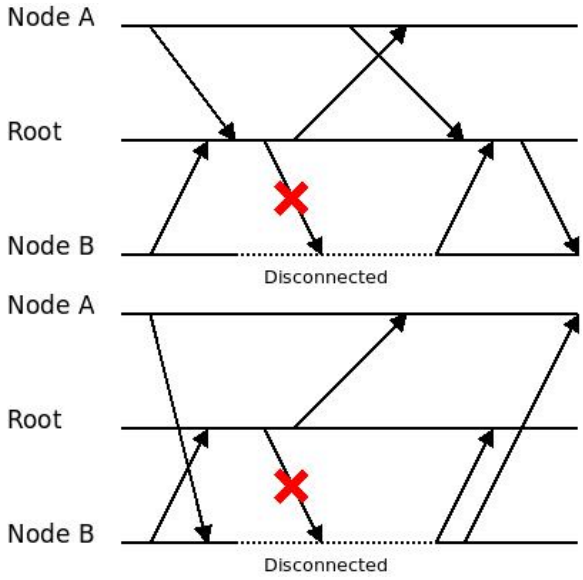


Figure 6.1: Web Delivery Model

As simple as the model might be, components in it could be vastly heterogenous if mapped to different techniques. Content stored in a node can either be web objects, SQL replies, codes or even entire databases. Communication in between can also be based on a variety of protocols.

As an online learning platform, users do not only passivly accept information, but also interact with Moodle through forum, personal blogs and quiz. All the changes made by users must be stored and seen everywhere. Thus, the system should not be read-only under any circumstance.

Moodle has been in service and adding new services should affect existing structure as less as possible. Also, steps of adapting changes should be properly designed to avoid crushing the service.

To maintain consistency and serve up-to-date content, a reasonable amount of communication overhead is necessary and is normally positive proportional to the extent of consistency. Although, due to the presumption of poor network connection and narrow bandwidth, different nodes in the system are preferably decoupled and autonomous.

The autonomy is also closely correlated to the ability of performing offline operation. Many distributed systems have the ability to detect and recover from network partitioning, although it normally leads to a compromise of consistensy and content freshness. When a user request a page, Moodle loads all previliges of the user, generate pages accordingly and log the session. This results in uncacheble content and interaction-must logins. It has been proven that consistency, high availability and partition tolerence are impossible to be achieved at same time[10][11], necessary trade-off has to be made

according to the condition and needs.

While the majority of web caching and content distribution techniques aim at better performance and delivery efficiency, we prioritize the ability of performing basic functionalities during network failure. We tolerate a relatively loose consistency while ensuring eventual convergence.

Lastly, to realize affordability, we mainly focus on open source techniques and free ware. Thankfully, many successful projects and tools have been made open source and publicly available. In the following sections of this chapter, we evaluate a variety of techniques against the criteria stated above and propose our solution based upon the conclusion. Several of potential solution are also tried out.

6.2 Push Web Service to Edge

6.2.1 Content Delivery Network

Content Delivery Network overlaps with Web Cache Proxy at the concept of pushing web content to users. A Content Delivery Network is a collaborative set of surrogate servers spanning the network, where web contents are mirrored[12]. Users will perceive a smaller latency while fetching content from a nearby CDN surrogate server rather than original web server. The essence of CDN is illustrated in Figure 6.2.

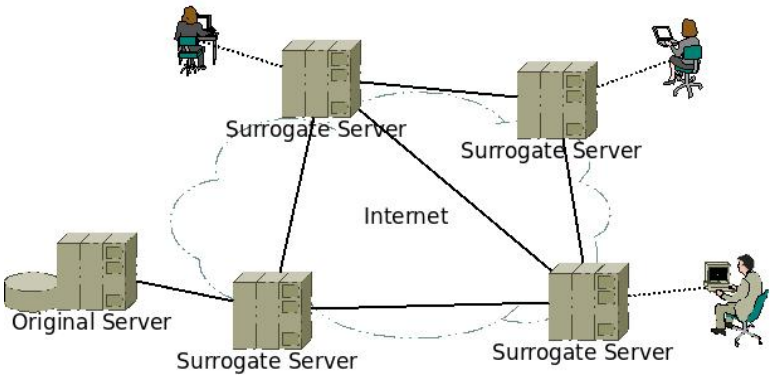


Figure 6.2: Content Distribution Network

Since more and more web services are evolving to provide dynamic content, CDN also takes advantages of cachebility hints when dealing with dynamic contents[13].

6.2.2 Simple Web Caching

An intuitive and common solution for the problem of limited bandwidth is to cache popular web content locally, as illustrated in Figure 6.3. A client-side

web cache proxy is typically deployed in user local network, requesting web servers on behalf of users, and cache web objects for further references. Web caching has been proven to be an effective approach to reduce bandwidth usage, user-perceived latency and loads on original server[14].

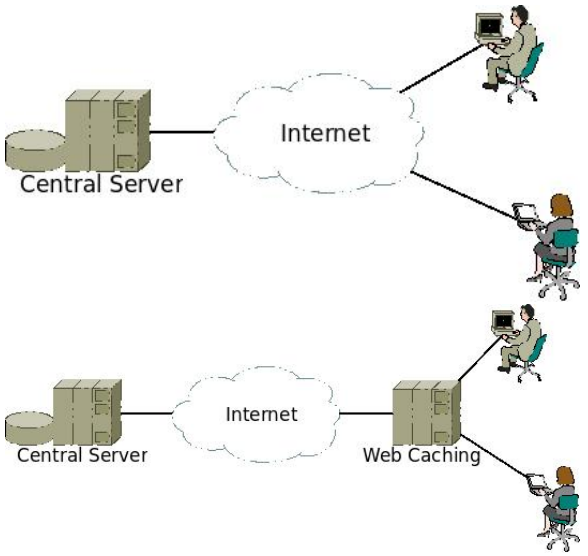


Figure 6.3: Serve users without and with a Gateway Cache

Web cache is greatly advantageous in our scenario that it does not require modification on web application, except that some TCP optimizations could be done between web cache proxy and web server frontend. Web cache proxy can continue serving requests with offline mode enabled, which also meets our requirements. The traffic between cache proxy and authentic server is standard HTTP request and response. The overhead and frequency of communication mainly depends on cache hit / cache miss ratio, expiration time and cache directory size.

Although, this solution encounters two major constraints in our case:

- Moodle pages are generated on-the-fly based on user information. Web cache cannot operate on its own during network failures, except for serving previously requested pages.
- Users are required to log in to browse. Thus, user-server interactions are always necessary, which also contradict with our purposes.

6.2.3 Page generation on Edge Server

To address the issue of dynamic content generation and client-server interaction, an intuitive and brute-force solution is to generate user-specific page at the edge. A comprehensive study of edge servers can be found in [12].

Four strategies are presented: (a) edge computing (b) content-aware caching (CAC) (c) content-blind caching (CBC) (d) data replication. In each of the strategy, the edge server attempts to reply user request on the behalf of original server with the information that is locally available. Edge computing still heavily relies on central database, hence out of our consideration. While CAC and CBC store partial database at the edge, data replication stores a complete copy of the database.

If the size and complexity of database permit, data replication is desired since it outperforms other strategies in both response speed and offline operation. Although, it adds another layer of complexity to perform transaction processing and maintain the consistency through mutliple distributed databases. Techniques to achieve a consistent distributed database system are presented in section 6.3.

Application code rarely changes in our case and is always one-way synchronized from original server, thus consistency can be relatively easy to achieve by periodically utilizing tools like rsync[15].

6.3 Multi-Master Database Synchronization

As addressed previously, consistency, availability and partitioned-network cannot be accomplished at the same time, according to CAP theorem[10]. Necessary trade-off has to be made to adapt to particular circumstance. Based on objectives defined previously, we prioritize availability and partitioned-network, while allowing loose consistency, as long as eventual consistency is guaranteed[16]. Pessimistic replication always guarantees consistent content perceived by users, thus widely adopted in distributed database system to enhance availability and performance. Although optimistic replication permits diverged databases, which is more suitable in our case. An exhaustive survey on optimistic replication is presented in [17].

Comprehensive theoretical researches have been made available although we find limited open source implementations that serve our purposes. We investigate several data replication technologies in this section based on following criteria:

- **Deployment over WAN**
  Most of distributed database system assumes a LAN environment, where delay and bandwidth do not evidently affect exchange of data among servers. Although in our case, databases are deployed over WAN, which is highly unreliable and network latency is not neglectable.

- **Serializibility**
  To achieve eventual consistency, concurrent changes made at different edge servers need to be serialized. The technology is responsible to extract relations in between and populate them through all nodes.

- **Conflict detection and reconsiliation**
  Conflict may occur while commiting concurrent changes that were made at edge servers. For example, two users may edit the same content, or one may delete one entry while another editing it. The technology should be capable of detecting and resolving these conflicts.

- **Minimum modification**
  In order to solve a practical problem with limited resources and budget, it is desired to rely on open source industrial level tools and extend it as less as possible.

6.3.1 Database Cluster

CouchDB[18] is an open source distributed database system developed in Apache. The most attractive feature of CouchDB is that it natively supports bi-directional synchronization among multiple database replicas and offline operations. When one replica is disconnected from the network, it retains autonomy and continues as a fully functional database from user point of view. Although, it fails to be our candidate since it is NoSQL database, whereas Moodle heavily relies on SQL calls and it is a significant task to modify Moodle to use NoSQL database.

MySQL Native Replication is shipped with most of MySQL standard distributions that provide built-in functionalities to replicate databases. The synchronization is uni-directional that databases on master node are replicated to slave nodes. ALthough bidirectional synchronization can be achieved by creating loop in the topology. A simplest 2-nodes example can be found in Figure X. Node A and node B synchronize with each other by maintaining mutual dominance. The figure also shows a more complicated topology where three nodes comprise a loop. Although, the synchronization can be easily broken by conflict and the reconciliations always require human intervention. To avoid insertion conflicts of auto-incremental keys, MySQL provides an option to increase incremental step. This feature is further discussed in section 6.3.4

MySQL Cluster[19] is an open source distributed database system based on MySQL. It supports database sharding and duplicating. A typical use case of MySQL Cluster is shown in Figure 6.5. Although, redundent copy of database can only be accessed with the presence of management master, and cannot be updated during network failures. Furthermore, database nodes are closely coupled with the assumption of LAN (low latency and high bandwidth).

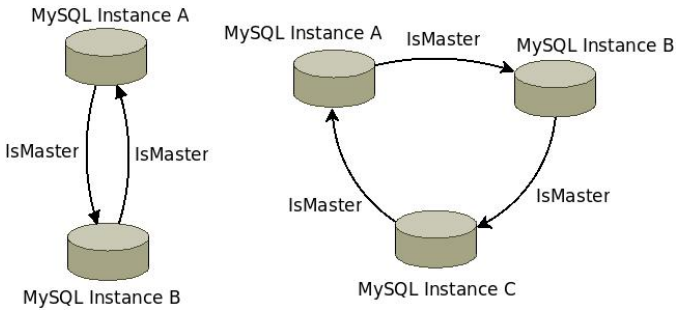


Figure 6.4: MySQL Replication

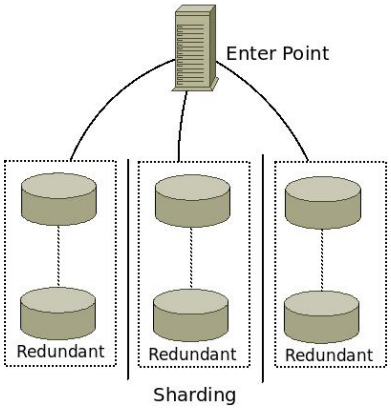


Figure 6.5: MySQL Cluster

6.3.2 Middleware-based Repliation System

Several studies also proposed the approaches to solve database synchronization at middleware-level[20][21][22]. C-JDBC [20] is an Java implementation of RAIDb[23], aiming at a framework to manage heterogenous databases. With built-in functionality of scheduling transaction processes, C-JDBC is perceived by users as a single virtual database. Although, the system is still centrally managed and could not handle partitioned network. Ganymed middleware system[22], inspired by C-JDBC, achieves consistency by serializing update/write requests at master and propogating changes to replicas in a lazy fashion. Users see a consistent data state (snapshot isolation), even though stale might it be. The limitation of these two middleware is also clear, that no write can be served during network failures.

6.3.3 Multi-Master Sycnhronization System

Similiar to MySQL Cluster Multi-Master setup, we found three state-of-the-art open source tools to the similiar end.

- **Galera**[24] is an open source synchronous database replication software developed to scale web application and provision high availability. It achieves consistency by optimistic locks and group communication. Galera is quorum-based and handles network partitioning by sacrificing the minority. Hence, Galera lacks the essencial features that we are seeking for and is out of consideration.

- **Tungsten**[25] replicates database asynchronously and allows loose consistency. Transactions are commited locally, and then propogated across all other nodes. Tungsten features offline operation and automatical recovery although it leave the responsibility of conflict avoidance completely to the application. Conflicts may disturbe replication and hard to trace.

- **SymmetricDS**[26] is another open source asynchronous database replication management tool. It has several built-in rules to detect and reconsile conflicts. It can be configured flexibly to meet different needs, for example, have different courses store in different sites. Also, one appealing feature is that SymmetricDS support file synchronization as well. Figure 6.6 birefly illustrates how SymmetricDS works and more details in terms of conflicts detection and reconsiliation are discussed in section 6.3.4

We can conclude from above comparison that SymmetricDS is closest to our critaria and can be properly extended to meet our requirements. In the following section, we explore SymmetricDS in details and propose an approach to extend it.

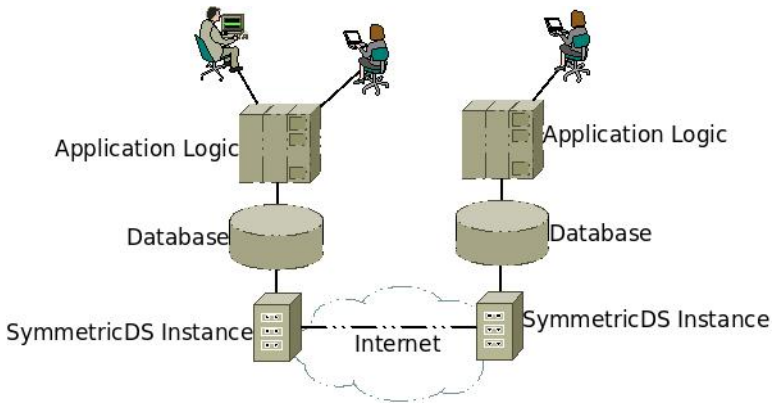


Figure 6.6: SymmetricDS

6.3.4 Conflicts Avoidance and Reconciliation

Before we start illustrating our approaches for conflict detection and resolution, several reasonable application-specific assumptions need to be made:

- No user will attempt to login into two different server simutaneously, thus the content belong to this user will not be modified at two different sites.
- Administration-related content is always synchronized through top-down approach, that is, user creation, deletion and modification are always synchronized in one-way.
- No local user will be able to reply to a post that is not propogated to this edge server yet.
- Volatile entries are not synchronized, such as cache and session.
- Deletion wins. And events that causally follow the deletion also get deleted. If users reply to a post on a remote server whereas the post gets deleted on central server, all replies will get deleted once converged.

As introduced in section 6.3.1, insertion conflicts of primary keys can be avoided by setting different incremental steps, as long as the primary key is auto-incremented integer, see Table 6.1.

Thankfully, all Moodle database tables are designed to use auto-incremental keys as primary key. By configuring incremental step and SymmetricDS, we are able to achieve a naive mutually consistent system. Configuring SymmetricDS is a nontrivial job and requires constant observation. Appendix A describes a work flow of configuring SymmetricDS.

Although, there are several major drawbacks of this design:

Primary Key	Node A	Node B	Node C
1	inseted by A		
2		inserted by B	
3			inserted by C
4	inserted by A		
5		inserted by B	

Table 6.1: MySQL instances with different auto-incremental steps

- It is not scalable. The incremental step limits the max number of servers in the system.
- It is blind to application. It does not check logical correctness of modification according to application, e.g. replies to a previously deleted thread can still be inserted into database.

To attack these problems, we examine the way SymmetricDS handles conflicts and propose an extension that is more flexible and tunable. SymmetricDS applies the concept of CVS (Concurrent Versions System) to detect conflicts at a granularity of table entry level. When a node propogates a change on one table entry, it sends the before-state of that entry along with the change. Upon receipt, remote node compares the current state of this entry and the history. If they differ, a conflict is detected. Even though conflict resolution is highly application-specific, SymmetricDS provides limited rules to resolve known types of conflicts.

A SymmetricDS node can be configured to either actively pull from other nodes, or passively waiting for changes being pushed. While optimistic replications normally assume a negotiation phase while attempting a resolution, push and pull in SymmetricDS are independent from each other. Thus, conflicts resolution can be sometime confusing. For example, suppose a scenario in Figure 6.7. Slave node is configured to push data to master while pulling changes from it. Since two operations are independent from each other, two replies to the same post are swapped, although the intention is to simply stack over.

To attain serializibility, several repliation systems applies a centralized algorithm that serialize all changes at master node and then propogate to slaves. Although, it often requires a closely connected system and can suffer from latency and disconnection. Inspired by Git and Operational Transformation [27], we propose a distributed work flow in Figure 6.8 where conflicts are resolved at edge server, similiar to the mechanism used in Dropbox[28].

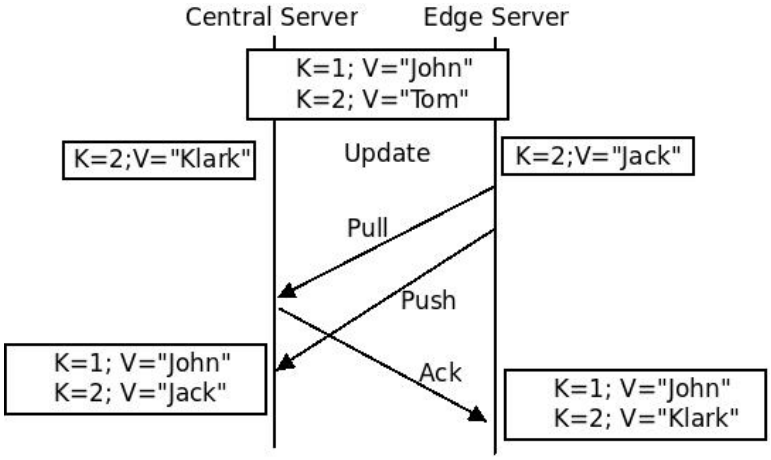


Figure 6.7: Misbahavior of SymmetricDS

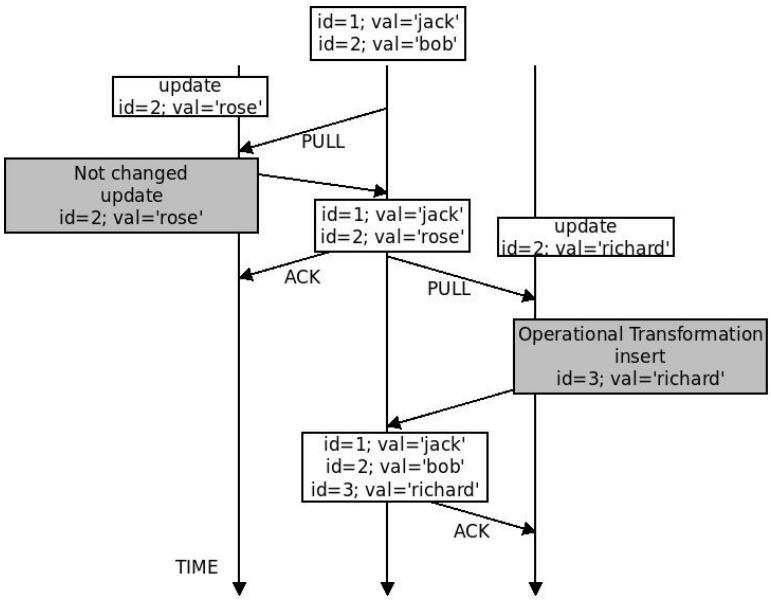


Figure 6.8: Distributed Conflict Resolution

Chapter 7

Low Power, yet Powerful

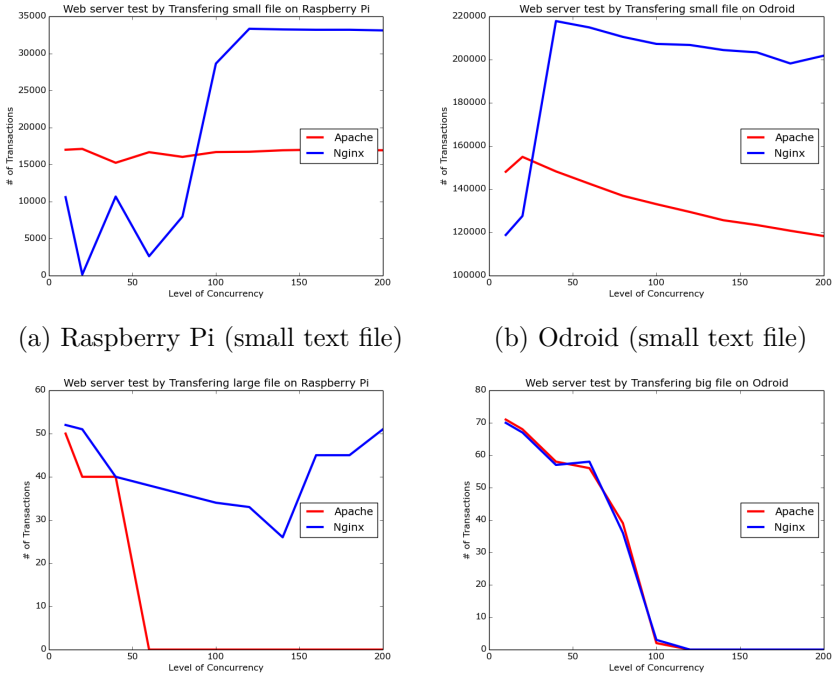
The ARM-based processors have received great attention for its characteristic of low power consumption and energy efficiency, especially in smart phone and portable device industry, where power consumption is one the most critical specifications. Furthermore, there is trend in server industry to shift to ARM-based architecture in order to cut off the bill of electricity. ARM is also ambitious in this area and about to publicize processors capable of virtualization. When it comes to rural development, power shortage has been forcing researchers and engineers to seek for alternative power sources. Lower power consumption of the equipments implies a bigger potential of surviving severe environment. Currently, we are exploring the possibilities of two platforms, namely Raspberry Pi and Odroid. The specifications of these two platform can be found in Table X In the following sections, we benchmark a variety of attributes of Odroid and Raspberry Pi. The objective is to clarify the capacity of these two platforms and an optimum form to run the web service. We test every component of a complete Moodle installation to identify the bottleneck of the application. Based on the results, we propose the optimum cluster solution that can be easily scaled out to serve more users.

7.1 Benchmark of Web Components

To test components of a complete Moodle installation, we design a expirement in Figure X. Each part is respectively substituted with either Odroid or Raspberry Pi, and remaining parts are running on a high-end machine which is much more powerful than these two platforms, in order to put enough siege on the testing target. Furthermore, simulated requests are generated from high-end machine as well.

7.1.1 Web Frontend

Most of the websites today are powered by Apache due to its long history and abundant extensions. Although Apache relies on a processed-based manner to handle new connection, which has limited the scalability and concurrency. Nginx is an event-based reverse proxy that handles request asynchronizily. It addresses C10K problem from the beginning and focus on scalability. To determine which server runs better on a resource constraint platform, we benchmark Nginx and Apache on both platform to evaluate the throughput, level of concurrency and response delay. We use siege to generate workload and compare the performance of web server with two different type of object: small text file and large JPG file. We siege the server in following setting:



(a) Raspberry Pi (small text file)

(b) Odroid (small text file)

(c) Raspberry Pi (big jpg file)

(d) Odroid (big jpg file)

Figure 7.1: A Benchmark of static file request on both platforms

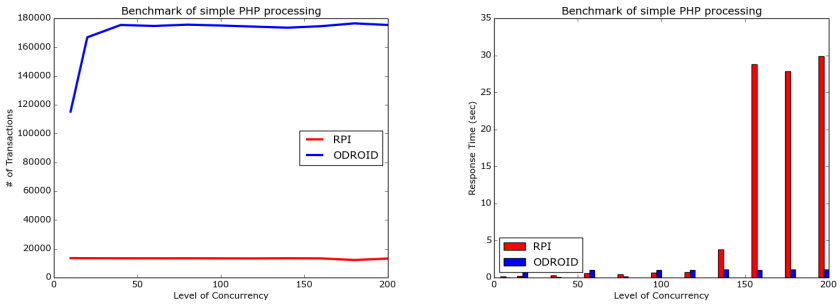
As shown if Figure 7.1, Nginx and Apache achieve comparable performance under a low level of concurrency, although Nginx outperforms Apache notebly when concurrency level increases. To be noticed, both web servers perform indistinguishably while serving large file. We observe fully occupied CPU in this specific test, which is due to intensive OS kernel processing of socket manipulation and packet tranfer. The impact of web server on the CPU is neglectable in this circumstance. Thus, expirement result in Figure

7.1(d) does not denote same performance of web servers.

7.1.2 PHP Processing

As a large PHP application, Moodle requires significant computational resources for PHP processing. Thus, a testbed is formed to benchmark PHP processing capacity of two platforms, see Figure X The inputs are two different PHP scripts:

- a php script simply echoes `Hello, world!`
- Moodle index page which requires heavy php processing.



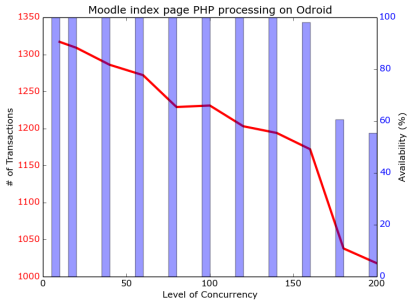
(a) $\sum trans$ (b) Longest Response Time

Figure 7.2: A Benchmark of simple PHP processing on both platforms

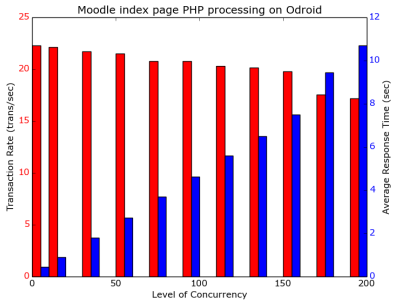
Figure 7.2(a) shows that Odroid can handle much more PHP request than Raspberry Pi. Overall response time is shorter for Odroid. Although, half of the CPU usage is taken by OS kernel during these two test and the difference satisfactorily convincing. The gap between two platforms is more evident when tested with heavier PHP processing, as shown in Figure 7.3.[1]

As shown in Figure7.3(a), transaction rate and availability drops under 180 concurrent requests occur. The latency is beyond tolerable under 50 concurrent requests, according to a study of tolerable waiting time of website**??**. As the total transaction number in this test is significantly lower than the previous one in Figure 7.2, we suspect that PHP processing power is the bottleneck in a moodle installation rather than database query and web frontend. Thus, we test the capacity of a standalone installation of Moodle, in which all components are in one box (Odroid), shown in Figure 7.4.

[1]We observe a 6 7 seconds delay to process index page of Moodle on Raspberry Pi and it can barely tolerate the mildest siege. Thus, the test result of Raspberry Pi is left out in this test.

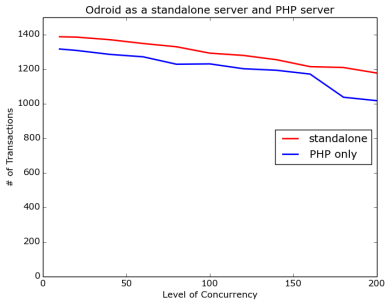


(a) $\sum trans$ & Availability

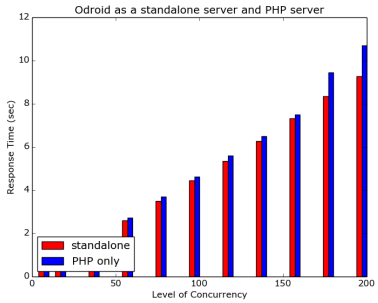


(b) $Rate_{trans}$ & T_{resp}

Figure 7.3: Moodle index processing on Odroid



(a) $\sum trans$ & Availability



(b) T_{resp}

Figure 7.4: Odroid as a standalone server and PHP server

Impact of adding web frontend does not evidently influence transaction rate, which implies that PHP processing capacity is the limiting factor in one Moodle installation.

7.1.3 Database

We apply the same method to test database although fail to exhaust the resource on Odroid before running out CPU and memory on our testbed high-end machine, hence we conclud that database query does not limit a Moodle installation before expanding PHP capacity.

7.2 Scale Out to Eliminate Bottleneck

To benefit most from currently available hardwares, we propose to form a small scale cluster than can be easily scaled out to cope with an increase of users. Furthermore, we investigate multiple different formations to find out the most economical setting. An intuitive solution is to put different

services in physically separated hardwares, as a structure shown in Figure 7.5. Nginx server in Rapsberry Pi features both web server and load balancer with upstream PHP servers running in Odroid. To cope with the increase of users, this setting can be easily scaled out by simply adding more hardware for PHP processing. Although, as we compare the Siege test result with a

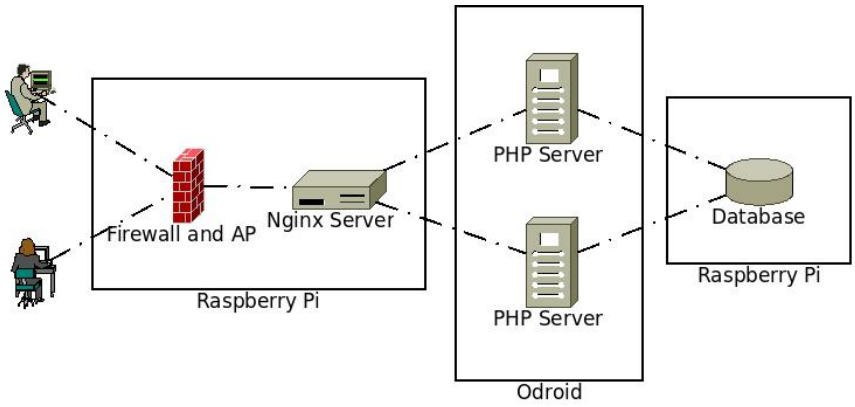


Figure 7.5: A naive form of cluster

standalone installation where all services run in one Odroid, we encounter a slightly lower performance, which is even worsen after we add one more PHP server, see Figure 7.6.

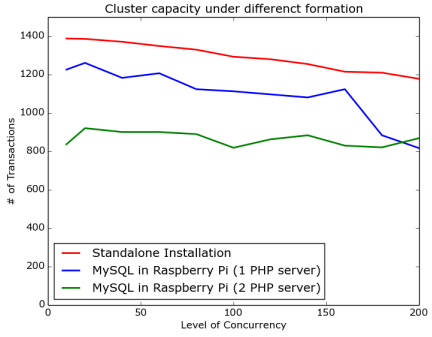
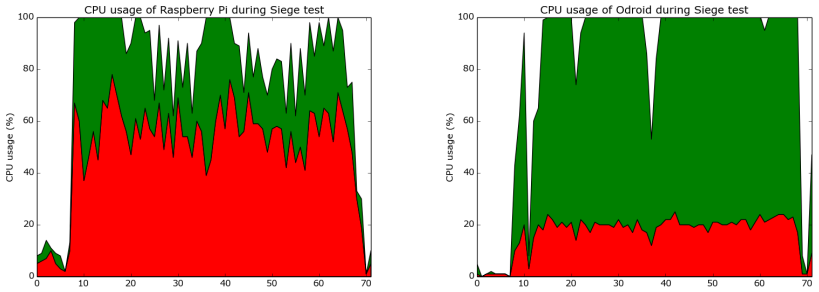


Figure 7.6: The Performance of running Database in Raspberry Pi

To further identify the cause, we siege the cluster with 100 simulated concurrent clietns and observe CPU usage of servers (CPU of Nginx server is far from full load hence left out in the result), the result is shown in Figure 7.7. We deduce that most of CPU resource in database server is consumed by OS kernel to handle concurrent sockets.

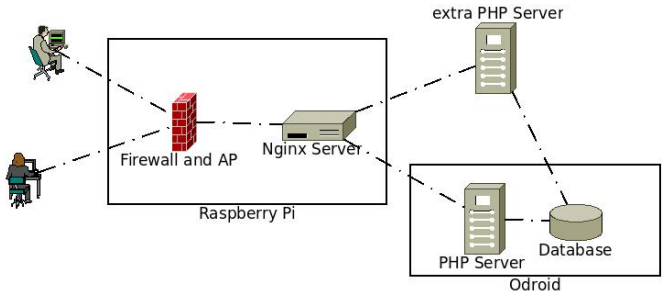
Hence, database needs to be more closely coupled with PHP server, which results in the cluster formation in Figure 7.8a. And the result meets our assumption that server capacity improves by adding extra PHP server,



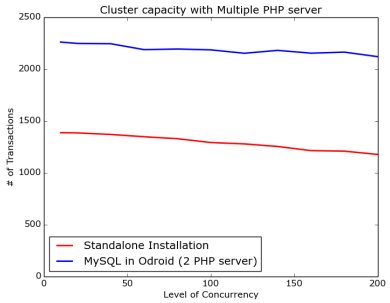
(a) database (b) PHP

Figure 7.7: CPU usage of servers

see Figure 7.8b



(a) A better form of cluster



(b) Performance of cluster with multiple PHP servers

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Starting from the user case, by applying the concept of divide-and-conquer, we are able to formulate the problem and identify the key challenges. Given the impossibility of achieving consistency and availability at the same time, we propose a multi-master system with reasonable assumptions and compromises. We surveyed a variety of existing technologies and investigate the potential to map them into our case. With extensions and proper configurations, SymmetricDS is tuned to serve our purpose. To achieve low-power consumption and affordability, we test the capacity and proposed a cluster which can be easily scaled out. We build Moodle web service on the cluster with integrated synchronization functionalities. We conclude that this prototype has the potential to serve in rural areas where highly available web services are desired and number of users is limited.

8.2 Future Work

At the beginning, the plan was to implement the system and test it in production environment. Although enormous effort has been put into exploring a variety of technologies. Also the overhead to adapt to local work culture disturbs original plan. Thus, the next step will be testing and debugging the system in a real production service.

All web applications encounter same obstacles when adapted to rural area. From the beginning of design, we aim at a generic solution for database-driven web services. This possibility should be further explored.

Bibliography

[1] K. Matthee, G. Mweemba, A. Pais, G. Van Stam, and M. Rijken, "Bringing internet connectivity to rural zambia using a collaborative approach," in *Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on.* IEEE, 2007, pp. 1–12.

[2] M. Pun, R. Shields, R. Poudel, and P. Mucci, "Nepal wireless networking project: case study and evaluation report," *Retrieved on October*, vol. 1, p. 2007, 2006.

[3] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[4] A. Nungu, T. Brown, and B. Pehrson, "Business model for developing world municipal broadband network-a case study," in *Global Information Infrastructure Symposium (GIIS), 2011.* IEEE, 2011, pp. 1–7.

[5] T. Jager *et al.*, *Soils of the Serengeti woodlands, Tanzania.* Pudoc Wageningen, 1982.

[6] http://www.out.ac.tz/. [Online]. Available: http://www.out.ac.tz/

[7] https://moodle.org/. [Online]. Available: https://moodle.org/

[8] http://www.aosabook.org/en/moodle.html. [Online]. Available: http://www.aosabook.org/en/moodle.html

[9] https://docs.moodle.org/. [Online]. Available: https://docs.moodle.org

[10] E. A. Brewer, "Towards robust distributed systems," in *PODC*, 2000, p. 7.

[11] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002.

[12] M. Pathan, R. Buyya, and A. Vakali, "Content delivery networks: State of the art, insights, and imperatives," in *Content Delivery Networks*. Springer, 2008, pp. 3–32.

[13] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *Internet Computing, IEEE*, vol. 6, no. 5, pp. 50–58, 2002.

[14] B. D. Davison, "A web caching primer," *Internet Computing, IEEE*, vol. 5, no. 4, pp. 38–45, 2001.

[15] A. Tridgell, *Efficient algorithms for sorting and synchronization*. Australian National University Canberra, 1999.

[16] W. Vogels, "Eventually consistent," *Communications of the ACM*, vol. 52, no. 1, pp. 40–44, 2009.

[17] Y. Saito and M. Shapiro, "Optimistic replication," *ACM Computing Surveys (CSUR)*, vol. 37, no. 1, pp. 42–81, 2005.

[18] http://couchdb.apache.org/. [Online]. Available: http://couchdb.apache.org/

[19] http://www.mysql.com/products/cluster/. [Online]. Available: http://www.mysql.com/products/cluster/

[20] E. Cecchet, "C-jdbc: a middleware framework for database clustering."

[21] C. Amza, A. L. Cox, and W. Zwaenepoel, "Conflict-aware scheduling for dynamic content applications."

[22] C. Plattner and G. Alonso, "Ganymed: Scalable replication for transactional web applications," in *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. Springer-Verlag New York, Inc., 2004, pp. 155–174.

[23] E. Cecchet, "Raidb: Redundant array of inexpensive databases," in *Parallel and Distributed Processing and Applications*. Springer, 2005, pp. 115–125.

[24] http://galeracluster.com/. [Online]. Available: http://galeracluster.com/

[25] http://www.continuent.com/solutions/clustering. [Online]. Available: http://www.continuent.com/solutions/clustering

[26] http://www.symmetricds.org/. [Online]. Available: http://www.symmetricds.org/

[27] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems," in *ACM SIGMOD Record*, vol. 18, no. 2. ACM, 1989, pp. 399–407.

[28] "Dropbox conflicts resolution," https://www.dropbox.com/developers/blog/48/how-the-datastore-api-handles-conflicts-part-1-basics-of-offline-conflict-handling.