

Kwaabana: File sharing for rural networks

David L. Johnson*
Meraka Institute, CSIR
Pretoria, South Africa
djohnson@csir.co.za

Elizabeth M. Belding
University of California
Santa Barbara
ebelding@cs.ucsb.edu

Consider Mudenda
Linknet, Macha Works
Macha, Zambia
consider.mudenda@machaworks.org

ABSTRACT

As usage of online social networks and social media continues to experience exponential growth, the amount of data being shared between users is increasing without bound. While this has revolutionized communication for many Internet users, users in rural or developing areas connected behind slow, congested gateways are falling increasingly behind the growing bandwidth requirements of the modern Internet. To address this problem, we present Kwaabana, a system to enable efficient sharing of content between users within a given region and to support reliable sharing of content between local village users and external users on the Internet. In this work, we describe the Kwaabana architecture and evaluate its performance in the rural village of Macha, Zambia. The results show that our localized file sharing service facilitates reliable sharing amongst rural users. Importantly, it also removes the cost barrier present for similar Internet-based services. We outline the process used by Kwaabana to achieve eventual database consistency and minimize impact on the Internet gateway link when synchronizing content between local and remote servers. In addition, we discuss some of the challenges specific to designing a solution for a remote community, and how we addressed those challenges.

Categories and Subject Descriptors

C.2.2 [Computer-Communications Networks]: Network Protocols-Applications; C.4 [Performance of Systems]: Performance attributes

General Terms

Measurement, Performance

Keywords

Rural network, Social network, Traffic analysis, Developing regions, Localization

*This work was completed while David was a Ph.D. student at UC Santa Barbara

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM DEV 4, 6–7 December 2013, Cape Town, South Africa
Copyright 2013 ACM 978-1-4503-2558-5/13/12 \$15.00
DOI: 10.1145/2537052.2537055.

1. INTRODUCTION

It is clear that online social networks (OSNs) have revolutionized communication worldwide. The statistics are astounding: As of 2012, 1.2 billion people use Facebook.¹ Twitter adds about 500,000 new users a day.² 800 million unique visits to YouTube occur every month.³ Our previous work demonstrated that even residents of the far reaches of rural Africa use Facebook frequently to communicate with others in their community as well as with friends, family, and colleagues worldwide [8, 7]. In these areas, online social networks (OSNs) are critical as alternate communication infrastructures (e.g. fixed telephone lines, cellular service) are infrequently available.

However, what is also clear is that access to OSNs is not uniformly available to the world's citizens. In developing regions, Internet access is often slow and unreliable; it can be orders of magnitude slower than in developed countries, as residents access the Internet through low bandwidth satellite or other long distance wireless links. This is coupled with the growth in size and complexity of the world wide web – the average web page size in 2011 was 48 times larger than the average size in 1995 (14.12K in 1995 [3] and 679K in 2011 [4]). Websites have also become more dynamic, and as a result, less cachable. This is particularly true of social media sites, where content can change at sub-minute granularity.

These challenges have resulted in a social media digital divide, where Internet users in developing regions with poor Internet access have difficulty actively participating in and contributing to social networks. Specifically, our prior work in Macha, Zambia showed that it is common for 75% of uploads to fail [8], making it difficult for Machans to become producers of Internet content.

Despite these challenges, however, there is a clear desire for participation. Our Macha traffic analysis has shown that Facebook is the most popular URL request, accounting for 15% of all web requests during our two month measurement window in 2011; Twitter was the second most popular URL request, accounting for 11.5% of all requests [9]. As of 2013, Facebook users come from 213 countries worldwide⁴ while tweets have appeared from 229 countries⁵.

¹<http://www.statisticbrain.com/social-networking-statistics>

²<http://www.jeffbullas.com/2011/09/02/20-stunning-social-media-statistics>

³<http://www.statisticbrain.com/youtube-statistics>

⁴www.socialbakers.com/facebook-statistics/

⁵<http://aworldoftweets.frogdesign.com>

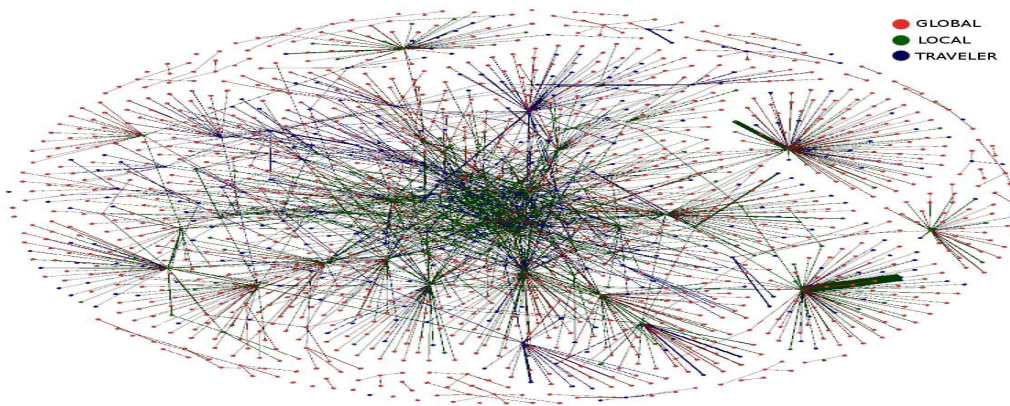


Figure 1: Social graph of Facebook instant message exchanges. Nodes represent users with color indicating the location. Edges represent messages sent between users with green representing messages between local users and red representing messages between local and global users. The thickness of the edge correlates to the number of sent messages.

Our prior work has studied the spatial locality of Facebook instant message (IM) exchanges in Macha and discovered that the majority of instant messaging and image sharing is between local users – users within a single village [9]. Other studies have shown similar findings, that as the Internet continues to grow, it is becoming “more local” [19, 14].

In this paper, we endeavor to take advantage of the spatial locality of social network traffic to provide a better content sharing experience to users in developing regions. Specifically, we develop and deploy Kwaabana, an architecture for optimizing local sharing of OSN content, while simultaneously facilitating streamlined content sharing with external users.

In our previous work [8], we created a Facebook application to allow users to share content locally in the village by storing content on a locally hosted Facebook application server. However, due to the high per-byte fees to access the Internet, we discovered that many users were excluded from using the service. This led us to an alternate solution — Kwaabana — that utilizes a local cloud storage solution, providing file hosting and sharing within the village without the hard requirement of an Internet connection; the Internet gateway is avoided when content is shared within the village, saving the gateway link for external traffic. A global Kwaabana server is put in place to allow village users to share content with non-local, i.e. “international”, users. Databases between these two servers are partially synchronized based on which content is shared between local and international users. A sophisticated synchronization engine is put in place to minimize disruption to interactive traffic, i.e. HTTP traffic. Synchronizations occur primarily when there is bandwidth available on the Internet gateway link.

We have deployed Kwaabana in Macha, Zambia and have continued to refine the system design based on community participation and feedback. We found that users utilize the service both as a means of locally backing up images from their mobile phones as well as for sharing images, documents, music and videos. Kwaabana has also been used to share large videos by disseminating a broadcast URL within the community Facebook group. Due to our smart syn-

chronization strategy, synchronization of files between the local and global Kwaabana servers shows little effect on web browsing performance in Macha; periods of low Internet gateway utilization are exploited for synchronization. We believe that Kwaabana will be key to enabling vibrant collaboration amongst users in remote communities, as well as sharing of content generated by residents of remote regions with the international community.

2. BACKGROUND

In our prior work, we collected and analysed network traces from the wireless network in Macha, Zambia over 2 months during February and March 2011 [9]. We discovered a number of interesting statistics about network usage, including a high degree of locality of interest in Facebook instant message (IM) chat: 54% of all sent messages were between local users even though only 35% of the user IDs captured by the trace were local to the village. We also established that content generation is severely hampered by poor network performance during peak usage times.

In the ensuing months we have continued to capture traffic in the Macha network. In April 2011, the network was upgraded from a slow 512kbps satellite access link to a dedicated 802.11 2Mbps link. In our subsequent analysis, we have discovered that the locality of interest trend is actually increasing. Between May and August 2011, the number of Facebook chat messages between local users increased to 60% while the number of local users dropped to 30%. The social graph capturing this trend is shown in Figure 1. In this graph, we place an edge between users who exchange at least one instant message; the thickness of the line correlates with the number of instant message exchanges measured in the trace. The colour of the edge represents the locality of the conversation; green edges for conversations between two local users and red edges for conversations between a local and external user. From this graph (and a supporting statistical analysis) it is clear that there are a large fraction of thick edges between local nodes, shown in green. The very noticeable thick edge in the bottom right of the graph represents 3100 local messages between two local users in the 70 day measurement (an average of 44 messages per day between the two local users).

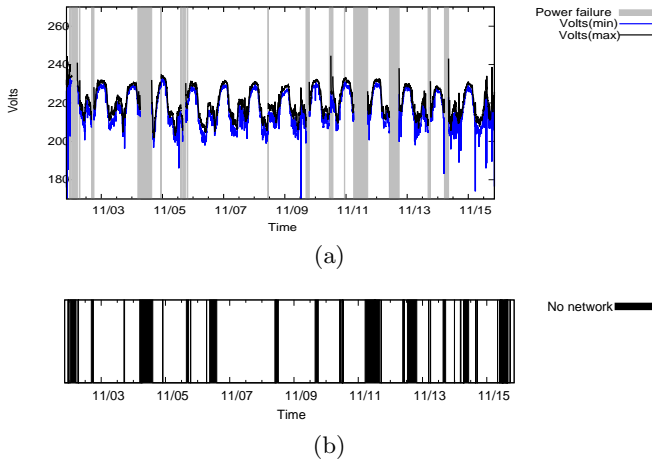


Figure 2: Power quality and availability in Macha over a 2 week period. (a) Gaps in the voltage reading depict the power failures. Dips below 200V represent brown outs. (b) The shaded regions show times when no Internet connectivity was available.

Also in [9], we found that locally generated Facebook images were three times more likely to be viewed by a local user than images generated by external users. This fact, together with the increase in IM locality, motivates our design goals for Kwaabana – to keep content shared between local users within the local network so that it can be served more quickly and successfully. This paper presents the first complete field-tested design of Kwaabana with a detailed discussion of the architecture and usage analysis.

2.1 Challenges in the environment

Installation and maintenance of ICT infrastructure in remote regions is a challenging endeavour [1, 13, 16, 17]. Problems range from lack of local suppliers to transport logistics. However, the lack of reliable grid power creates additional, and on-going, challenges.

To better understand our target environment, we monitored the grid power and network availability in Macha, Zambia over four weeks. During our measurement window, we found that Internet connectivity was not available to equipment approximately 40% of the time and power was not available approximately 15% of the time. The lack of Internet connectivity is often due to upstream problems in the grid or connectivity infrastructure in Zambia. It can also be due to the need for local equipment to be reset after becoming unstable during power failures or brown outs (voltage drops by more than 20% of the standard voltage, 220V). The measured voltage on a raw power outlet in Macha as well as the power failures during a sample two week period is shown in Figure 2(a). Multiple brown outs are clearly seen during the measurement period. These have the potential to put equipment in an unstable state. Internet availability is shown in Figure 2(b). It is clear that there is often no connectivity to the outside world even when power is available.

This lack of power or Internet connectivity must influence the design of our system. Specifically, we must pay careful attention to the need to resume file synchronization after

connectivity outages experienced by our server in order to ensure data transmission accuracy. The system should also continue to provide functionality to local users while Internet connectivity is not available due to upstream power failure issues.

3. ARCHITECTURE OF KWAABANA

The goal of Kwaabana is to optimize file sharing and communication between local users, minimizing use of the Internet access link. Files shared between local users should be stored and accessed locally, while files shared with a user external to the local network should use the Internet access link, but should minimize negative impact on interactive web browsing traffic. Kwaabana is installed on a server in a village for use by local users; we call this the “village Kwaabana server”. Kwaabana is also installed on a central, external server, in this case at UCSB, for use by all other users external to the village; we call this the “global Kwaabana server”. We have designed Kwaabana such that any number of village servers can exist, as we hope to spread our deployment to communities surrounding Macha, and beyond.

Our initial version of Kwaabana was built as a Facebook application because of Facebook’s prevalence in developing regions [8]. After deployment of the Facebook-based Kwaabana, we discovered that the integration with Facebook prevented some users from participating due to Internet access costs; many users cannot afford to pay Internet connectivity charges. When the user had to log in with their Facebook account, Facebook itself had to be contacted, resulting in per-byte usage fees from the local Internet provider. To provide broader access, we designed a second version of Kwaabana, this time integrated with an open source file hosting service called *ownCloud* [12].

ownCloud did not have a social networking component and files could be shared with anybody registered on *ownCloud*. In order to leverage the wide use of Facebook in the village, we added an *ownCloud* plugin which allows a user to import their Facebook friend list into *ownCloud*. A user associates their *ownCloud* account with their Facebook account. Friend relationships that exist in Facebook are mirrored in *ownCloud* once both users in the friend relationship have associated their Facebook accounts. These learned relationships are available in *ownCloud* even when the Internet link is down. Sharing of content is then limited to users in a friend list. Users also receive notifications on their Facebook account when content is shared with them on Kwaabana.

Kwaabana consists of a set of networking components integrated with a cloud storage application. The architecture of the system is shown in Figure 3. The primary networking components include a traffic interception system (*Interceptor*) to re-direct users to Kwaabana should they use file sharing Internet-services, and a *synchronization engine* that carries out lazy database and file synchronization. The *synchronization engine* maintains partial consistency between the local and global cloud storage server at UCSB while minimizing impact on interactive browsing traffic.

Content on the local and global cloud storage database is only partially synchronized; there is no need to have knowledge of an international user’s file if it has not been shared with a local village user, and vice versa. Once files are uploaded to Kwaabana, they are stored in the user’s personal storage space. The user can then share those files with other

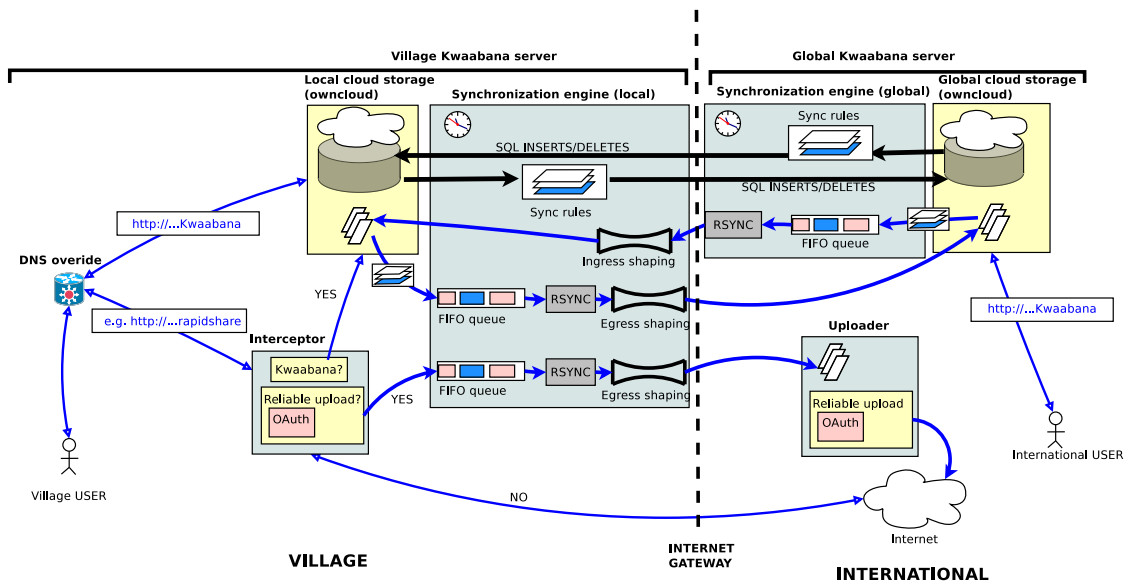


Figure 3: The Kwaabana architecture.

local or global users. If the files are shared between local users, the files are only stored on the local cloud storage server. Should a local user share a file with an external user, the file is placed in an upload *FIFO queue* to synchronize with the global cloud storage server. A set of *sync rules* are checked to decide which records in the local database need to be inserted into the global database. A scheduler executes the list of database updates at a regular interval and triggers the synchronization engine to upload files from the queue. We describe the database design in section 3.4.

We carry out file synchronization through a service that prioritizes interactive traffic such as HTTP. Our non-interactive file uploads and downloads receive a small slice of the bandwidth during periods of high utilization, and a larger slice when utilization is low. The service also uses the Linux *rsync* tool which is able to recover from partially synchronized files should there be severe congestion or a network or power failure. This service is critical as our prior work discovered that as many as 75% of file uploads are unsuccessful due to network congestion over the slow Internet access link [8]. We describe the synchronization service in section 3.5.

Kwaabana is available to all users, regardless of whether they reside in a rural region, via a single URL. However, communication between users within the local village is routed to the village Kwaabana service by the *DNS override* service that overrides the global Internet DNS entry. Rural users who attempt to upload files to common file hosting sites such as youtube are also intercepted to check whether the user intends to share this file with other local users. If the user does intend to share the file locally, they are redirected to the Kwaabana service to upload their file and share with a local user. We describe this file upload interception in section 3.2.

In the following sections, we describe how each of these components work together to provide the full set of Kwaabana functionality.

3.1 DNS override

It is common practice in the Internet to ensure that requests to servers are load balanced across replicated servers. In many cases these servers are replicated worldwide and the server that is closest to the client is used. Our Kwaabana server (www.village-net.org) is registered on a global DNS server to a machine in the USA. However, in the rural community the local DNS cache must be manually edited to point this domain to a local machine, redirecting all Kwaabana application requests to the local server. This reduces not only delay and Internet costs, but also simplifies deployment as local and external users only need to recall a single URL.

3.2 File upload interceptions

There are a number of monolithic Internet-based file sharing services, such as youtube and yousendit, that allow a user to share files. It is, however, a waste of costly Internet bandwidth to make use of these services to share content between local users in the village. To avoid this, uploads to these services are intercepted using an *iptables* filter and redirected to the Kwaabana service. The goal is to introduce users who may not be using Kwaabana to the service when they use Internet services that could be performed more efficiently with Kwaabana.

Once interception has occurred, the user is asked whether the file being uploaded is destined for a local user. If so, an option is presented to use the Kwaabana service. An additional option is presented to share files with outside users that allows the user to use a more reliable upload service. We have found in previous studies that uploads often time out due to network congestion or power failures [8], and that file hosting services do not allow users to recover from a failed upload. Use of a reliable upload service ensures that the file is first synchronized using a recoverable upload process to a server at UCSB running our uploader service. From there it

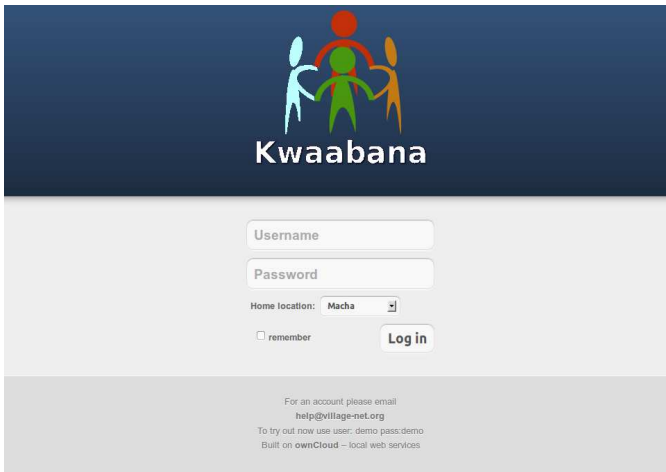


Figure 4: The main Kwaabana login page. Notice the option to select location.

is uploaded to a file sharing service. A user’s OAuth⁶ credentials, an authentication mechanism used by most file hosting services, are captured during the upload to re-authenticate with the file sharing service from our UCSB server. The file synchronization utilizes the Linux *rsync* tool, which is capable of recovering from a partial file upload. A file is only removed from the upload queue once it is successfully uploaded. When a server boots after a power failure, the synchronization service checks the upload queue and reruns the *rsync* tool on the first file in the FIFO queue, which may be partially uploaded or not uploaded at all.

3.3 User interface design

Our initial Facebook-based Kwaabana application provided a very simple upload and sharing interface which allowed users to share files with their Facebook contacts [8]. After we noticed no significant uptake of the service in the community due to exclusion of users without Internet vouchers and the lack of a rich interface that provided other features such as a media player, we modified the design to make use of a completely localized file storage and sharing solution that does not require the user to use the Internet connection.

The new Kwaabana file sharing service utilizes a local cloud storage solution. A user registers her name on the local service and then proceeds to upload and share content with other local or external registered users. The user interface is built on the ownCloud open source file hosting project. We incorporate our synchronization modifications and a feature to allow users to associate their account with their Facebook profile. This enables them to receive notifications on their Facebook account if a user shares content with them when they are logged into the Internet.

The Kwaabana login screen is shown in Figure 4. An account is requested from the administrator by sending an email request or, in the case of a user who does not have an Internet voucher, by physically visiting the administrator in the village. A user name and password is entered, followed by a location that is set to the current location by default. The purpose of the location entry is to allow visitors who

⁶Open standard for authorization.

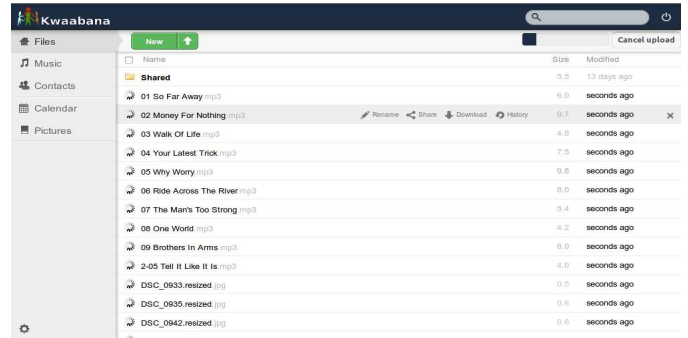


Figure 5: The Kwaabana file explorer page. Selecting “share” from this page allows users to share the file with users registered in the Kwaabana system.

have an account on a remote Kwaabana server to access their files.

The main Kwaabana file explorer interface is shown in Figure 5. The choices a user can make from this ownCloud interface are: (a) upload a new file or set of files. Files sizes up to 2GB are supported; (b) view supported media, such as pictures or videos. Pictures can be viewed as a slide show; (c) listen to their music collection directly from the web browser; (d) download files to their computer; or (e) share a file with another registered user. When sharing a file, a user can select whether the shared file is allowed to be edited. Files that have been shared with a user are placed in the *shared* folder.

When a user logs in to an external Kwaabana server (i.e. a server outside of her home community), a full set of file meta data is retrieved from the remote server; however, the actual files are not synchronized unless they are requested. After logging in the user is presented with a message: “Requesting your set of files from the remote Kwaabana server.” Once the file list is synchronized, the full set of files is shown in the file explorer interface. When selecting a file, a user is presented with an option to request a file from the remote server. The process of synchronizing the file list and file is presented in sections 3.4 and 3.5.

3.4 Database design

Kwaabana stores user and file information, as well as information on sharing between users, in a *mysql* database. Files are stored in a directory structure in which a file is found by traversing folders related to the owner and owner location. The structure of the database showing the core tables in the Kwaabana database as well as the directory structure is shown in Figure 6.⁷

To identify users, a user creates a unique user name with a location id attached to this user account. This information is stored in the *users* table. It is possible for two people with the same user name to register in different locations as the combination of user name and location id forms a primary key. As new locations are added they are stored in the *locations* table.

When files are uploaded by a user, the user automatically becomes the owner of these files. File information such as the file name, owner of the file, location and other file meta

⁷Not all fields and tables are shown. What is shown is sufficient to illustrate the core architecture of the system.

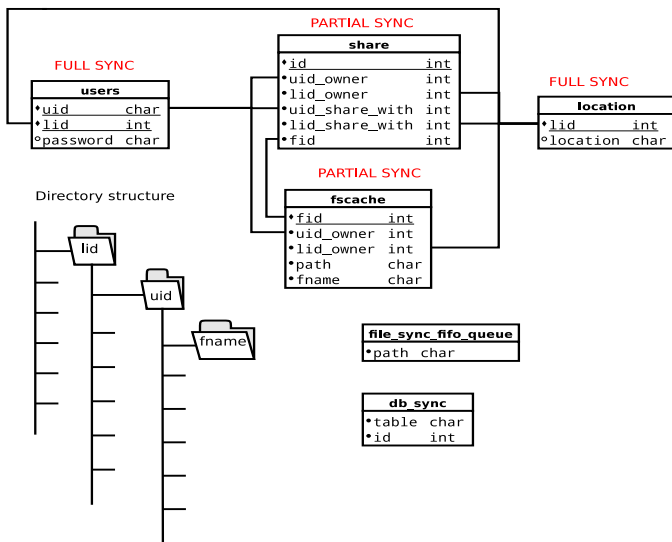


Figure 6: The database design, showing tables that support locality and file sharing across different Kwaabana locations.

data not shown, is stored in the *fscache* table. When a file is shared with another user, the owner id and owner location id, as well as the user id and location of the person with which the file is shared, is stored together with the file name in the *share* table.

Synchronization between databases at different locations is partial; it is not necessary to have knowledge of a set of files at another location if the files are not shared with user outside that location. However, the *users* table is fully synchronized in order for users at different sites to have knowledge of each other and share files. The *locations* table is also fully synchronized as new locations are added to the system. From the point of view of a local user, the *fscache* table is partially synchronized as it only needs to store information for files owned by local users and for files from remote users that have shared files with local users. The *share* table is also partially synchronized as it only needs to store information about files shared between local users and files shared between local and remote users; no knowledge of files shared between two remote users is required.

Suppose two local users share a file with each other. A local user, *Bob*, with location id 1 shares a file with id 11 with Alice at location id 1. A new entry will be placed in the *share* table. If the new record id in the *share* table is 9, the new entry in the *share* table is {9,Bob,1,Alice,1,11}.

Suppose a village user shares a file with an external user. A local user, *Bob*, with location id 1 shares a file with id 11 with Alice at location id 2. Again a record would be placed in the *share* table, in this case {9,Bob,1,Alice,2,11}. In addition the *db_sync* table will receive two entries describing the records from tables that need synchronization. The first entry would be {fscache,11}, the record from the *fscache* table describing the file needing synchronization. The second entry would be {share,9}, the record from the *share* table describing the share relationships of the file that needs synchronization.

As discussed in Section 3.3, it is also possible for a remote user logged in to a local server to request a copy of their files.

In this instance, all the ids for the complete set of records for the remote user in the *fscache* table are placed in the *db_sync* table.

A scheduler in the synchronization engine runs a task regularly that converts the entries in the *db_sync* table into a set of SQL INSERT statements stored in a file and sent to the remote database to be executed. Thus, only a partial set of relevant updates are executed on the remote database. Should users or files be deleted, a set of SQL DELETE statements will be created in a similar way. The bandwidth requirement for sharing these SQL update files is low as they are text updates that only reflect differential changes.

The actual files that are shared between local and remote users are entered into a fifo queue in table *file_sync_fifo_queue*. This table also holds files that are requested from a remote user as discussed in Section 3.3. A synchronization service, discussed in Section 3.5, is used to synchronize the files in this queue between the local and remote Kwaabana servers.

In the future we plan on adding support for multiple villages. In a multiple village scenario, we believe the current database design will be sufficient to support synchronization across multiple villages. However, the strategy used for synchronization between villages should preferably first synchronize to the global server and then rely on the remote villages, pulling updates from the global server due to the unreliability of power or connectivity in remote rural villages.

3.5 Synchronization and traffic shaping

The synchronization service is designed to synchronize files in the synchronization queue between the local Kwaabana file store and the global Kwaabana file store and vice versa. To optimize use of the limited-bandwidth Internet gateway that frequently exists in rural communities, we shape the synchronization traffic to minimize impact on the browsing activity of users while optimizing usage of the unused portion of bandwidth available in the Internet link.

The Traffic Control subsystem of the Linux Kernel is a sophisticated set of tools to shape traffic before it is passed to the network device. It allows the establishment of different queuing disciplines for packets based on classification according to criteria such as ports or IP addresses or any information available in the IP header (i.e. Type of Service (TOS)). These queuing disciplines, or *qdiscs*, represent the scheduling policies applied to a queue. Several types of *qdiscs* exist such as FIFO, Stochastic Fair Queuing (SFQ) and Token Bucket Filters (TBF). To shape the synchronization traffic, which is sent using the Linux *rsync* tool, we allocate a unique port to the synchronization process and use this to classify the synchronization traffic. As discussed in section 3.2, the combination of a FIFO queue, in which a file entry is only removed once the file synchronization is complete, and the recovery feature of the *rsync* tool make the system robust against upload failures due to power or network outages.

In order to shape traffic on the incoming (ingress) and outgoing (egress) links, we need to take control of the Linux queue attached to the network device. To do so, the Internet gateway machine needs to become the bottleneck for the traffic flowing to the Internet. In Macha, there is a 10 Mbps Ethernet link between the gateway machine and the WiFi router; the WiFi router provides a 2.1 Mbps⁸ link to the In-

⁸2.1Mbps is the SLA agreement with the ISP.

ternet. If the outgoing traffic is not shaped to just below 2.1 Mbps, all traffic will build up in a queue at the WiFi router which we cannot control. To control the queue, we shape the outgoing link to 2Mbps. We then classify the traffic based on any number of different criteria, such as HTTP traffic or traffic using the port allocated for file synchronization.

In a gateway serving a community where web traffic dominates, it is advantageous to divide the traffic so that most of the bandwidth is allocated to HTTP traffic. Traffic that needs low latency, such as TCP ACKs, should be pushed to the network device as quickly as possible. Because it is not interactive, synchronization traffic should receive a small slice of the bandwidth if other services are using the link, but it should consume as much as possible when the link is dormant. To split the traffic we use a hierarchical token bucket queuing discipline. This allows us to set a token bucket-based rate when the link is under contention but allows the service to borrow tokens from other classes when capacity is available, up to a ceiling rate. HTTP traffic is sent to a stochastic fair queue to avoid one HTTP connection dominating another. ACK packets are sent to a FIFO queue to allow them to reach the network interface as fast as possible. The Kwaabana traffic is also sent to a FIFO queue as only one file is synchronized at a time. The system design is shown in Figure 7.

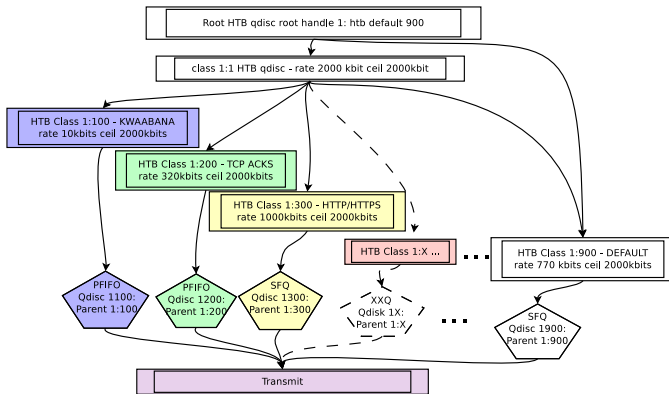


Figure 7: Traffic shaping design for Kwaabana using the Traffic Control subsystem of the Linux Kernel.

4. MACHA DEPLOYMENT

To evaluate Kwaabana, the application and synchronization engine was installed in the local network in Macha, Zambia. This network is managed by the Macha Works organization through the LinkNet project [10].

System operation and performance was assessed using both real user activity and simulated file sharing between a local and global user. Simulated sharing between local and global users was used to study the effect of synchronization traffic on interactive browsing traffic at the Internet gateway. After successful deployment, we monitored in-situ user activity to understand typical usage patterns in terms of sharing frequency and file types and sizes. We also wanted to investigate the user acceptance of a technology intervention in a specific cultural context. Macha shares much in common with other sub-Saharan countries in Africa, specifically an Ubuntu culture [16] in which community needs have priority over individual needs. Because of this vast difference

from western culture, we took a participatory approach by including local community members in the design and customization of the software. We describe this approach in section 4.1.

4.1 Introduction of Kwaabana

The introduction of new technology in a rural environment requires community involvement in the process of shaping the ultimate design of the system [15, 2]. Macha is an oral culture in which dissemination of ideas occurs chiefly through relationships and by word of mouth. Interventions, therefore, are not possible from a distance but require a presence in the community.

In order to introduce Kwaabana in Macha, we spent many months working with trusted community leaders to analyse usage patterns in the network and to assess user needs. This helped motivate an initial design to match the community requirements. This was followed by six weeks in the village customizing the design and graphic user interface through one-on-one consultations and workshops. An initial rural networking workshop, open to all local residents, was held in the community. An outdoor wall was used to screen a presentation explaining the network patterns in the community, including the locality of interest that motivated the design of Kwaabana. This was followed by a brief demonstration of the initial system.

Members of the community were then involved in choosing an appropriate local Tonga name. They selected “Kwaabana”, which means “share”. They also designed the Kwaabana logo, which to them represents African Ubuntu culture.

Once the system was deployed, we received constant feedback from users about features they would like to see added. For example, our initial system only supported uploads as large as 100 MB; we thought most people would share content created on their mobile phones with limited resolution and storage capacity. We soon discovered that users wanted to share high-resolution movies that were over 1 GB. In another example, our initial system required users to first log into Facebook in order to access Kwaabana. We had made this decision because of the high prevalence of Facebook, and because we thought users would want to continue to use a platform with which they were already familiar. However, we soon discovered that integration with Facebook using a simple application GUI excluded many users who could not afford the high cost of Internet access in the village (\$30 for a 1GB voucher) or who wanted a more media-rich file sharing experience. This motivated us to create a full off-line version of the service based on ownCloud which has a rich set of media features.

4.2 Online survey

To understand current sharing behaviour in the community, we launched an online survey. The survey asked questions about the type of media community members produce, such as audio, images or video. It then requested the respondents to report on their methods for sharing this media with their local or non-local contacts and obstacles they experienced sharing content. We used this survey to guide our design of Kwaabana to more fully facilitate content sharing.

Interview participation was on a voluntary basis and there were no material awards associated with the participation. The online survey was made freely available on the login

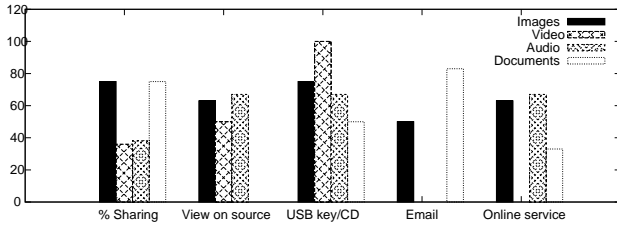


Figure 8: Results from online survey on current file sharing trends in Macha.

page of the Internet service provider. Some users completed the survey on their own accord while others were introduced to the survey by a trusted local community member.

A total of 15 users responded to the online survey in Macha. The participants’ age ranged from 18 to 35; 9 of them were female and 6 were male. The majority of users made use of mobile phones to produce content: 62% used phones to capture images, 60% used phones to capture video, and 40% for recording audio. For each type of media, we asked whether the user shared this content with others and what techniques were utilized for sharing. The results are shown in Figure 8. The overwhelming majority made use of external media, such as CDs and USB flash drives, to share content with others. All media was shared online to some extent except for video; video was shared exclusively via removable media. We probed this behaviour further by asking about obstacles to sharing content. 87% of the respondents selected cost (i.e. of Internet access due to per-byte usage fees) as a significant obstacle; 75% selected speed; and 50% selected reliability of the Internet as impediments to sharing.

From this survey it is clear that Kwaabana has great potential to increase content sharing and online collaboration within the community. Video sharing clearly has the most to gain from a local file sharing service.

5. EVALUATION

In this section we evaluate the operation and some post-deployment restricted usage of Kwaabana. We study the type of content uploaded and shared on Kwaabana. We depict the sharing between users as a social graph to understand the distribution of sharing behaviour across the user base. We compare this social graph to previous work on the locality of interest for Facebook images and instant-messages. We also evaluate the synchronization component of Kwaabana using simulated file shares between local and remote users. We carry out experiments during normal peak usage times in the network to ensure that our synchronization component has minimal effect on normal interactive web browsing.

5.1 Evaluation of shared content for real users

Following our deployment in Macha, we provided access to Kwaabana to a limited number of users to test its operation “in the wild”. For the evaluation described in this section, we disabled synchronization between the local and global system until Kwaabana could be integrated with the voucher based billing system. This evaluation, therefore, only presents local sharing behaviour within Macha. During these initial tests, there were 12 registered users who were

encouraged to explore the use of Kwaabana for file sharing. The results are from the initial two weeks of usage following deployment.

Table 1: Statistics for files stored on Kwaabana.

| File Type | Quantity | Total Size(MB) |
|------------|----------|----------------|
| image | 51(72%) | 169(18%) |
| audio | 5(7%) | 71(8%) |
| pdf | 12(17%) | 6(4%) |
| video | 2(3%) | 669(73%) |
| executable | 1(1%) | 3(0.3%) |

Table 2: Statistics for files shared on Kwaabana.

| File Type | Unique files | Total shared | Share ratio |
|------------|--------------|--------------|-------------|
| image | 6(43%) | 22(46%) | 3.6:1 |
| audio | 4(29%) | 13(27%) | 3.3:1 |
| pdf | 3(21%) | 11(23%) | 3.6:1 |
| video | 1(7%) | 2(4%) | 2:1 |
| executable | 0(0%) | 0(0%) | 0:0 |

Table 1 shows a summary of the quantity of files uploaded over two weeks using the Kwaabana service, while table 2 summarizes the quantity of files shared. Images are the dominant media type; 72% of the total files uploaded to the server. However, after only two videos were uploaded to the server, 73% of the storage space was consumed. Only 6 of the 51 stored images were shared with other users whereas all the videos were shared with other users; one of the videos was shared with users registered on Kwaabana and the other video was broadcast to all users in Macha by posting a URL link to the video on a Facebook page. In the audio category 4 out of the 5 audio files, which were mp3 music tracks, were shared with other users. Our results indicate that users have different intentions when uploading content to Kwaabana. When content is generated by the user – as most images are – Kwaabana is used as a backup service for this content with a small fraction of these files shared with other users. Recall that using a Internet file hosting service for backup would be very costly and slow in a remote rural area like Macha. When the content is not locally generated, in the case of downloaded music or videos, the content stored on Kwaabana is uploaded with a clear intention to share this with other users as backup of these files is not as critical.

We define the share ratio of the content as an average measure of the number of files shared with other users for a specific file type, calculated simply by dividing the total number of shared instances of a file type by the total number of unique files of that type. We find that the share ratios are fairly similar for images, audio, and pdf files – the share ratio of each is greater than 3:1. Our previous work, which analyzed locality of interest in Facebook images [8], showed that local content was viewed 5.2 times on average by local users, compared to externally generated content, which was only viewed 1.5 times on average. We believe the share ratios are likely increase further as usage of Kwaabana grows, particularly as it grows among users without access to Facebook.

We captured an event where one of the large 600 MB videos uploaded to Kwaabana was broadcast to all users in

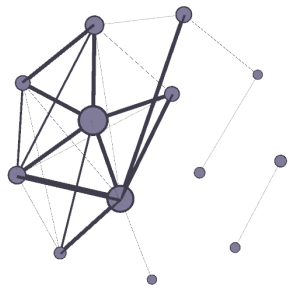


Figure 9: Social graph of Kwaabana local file sharing in Macha. Node sizes indicate the weighted node degree and edge thickness depicts the number of files shared between users.

Macha using a Facebook link to the content. Analyzing the apache access logs revealed that this video was viewed three times within the first day of posting. Although the video was not directly shown to users of Kwaabana, the broadcast of this video achieved a share ratio of 3:1 within the first day and saved 600 MB of upload bandwidth and 1.8 GB of download bandwidth on the Internet gateway. This is a financial saving of \$72, given that 1 GB vouchers cost \$30.

The social graph depicting sharing behaviour between users is shown in Figure 9. Node sizes indicate the weighted node degree and edges depict the number of files shared between users. It is clear from this graph that a few dominant users share the majority of files and many users only share a few files. This behaviour mimics social graph patterns we have seen in Facebook instant messaging [9]. There are a few users who act as ‘information conduits’ or ‘information generators’ and a large number of other users who are consumers of this content. These are early results and, considering the small size of the community testing the system, it is hard to make broad conclusions; however, the results demonstrate that local services should mimic behaviour seen in non-local services.

5.2 Evaluation of synchronization

To evaluate the synchronization strategy using the Linux traffic control system, we monitored the downlink of the Internet gateway during a simulated file synchronization between the global and local Kwaabana sever; this effectively simulates an international user sharing a file with a local village user. Figure 10 shows a trace of synchronization traffic and all other Internet traffic passing through the gateway; samples of the downlink throughput were taken every 2 seconds.

One 15 MB file was placed in the upload queue (from the point of view of the global Kwaabana server). The incoming queue on the village server was restricted to 1.2 Mbps using a hierarchical token bucket (HTB). As can be seen from the plot, the HTB allows some momentary throughput to reach 2 Mbps; however, over a larger averaging window, the downlink throughput is 1.2 Mbps. The scheduler in the synchronization engine started the download at 13:47:30. At this stage the browsing traffic averaged around 800 kbps, and the synchronization service took full advantage of the spare capacity of about 400 kbps by borrowing tokens from the other *qdisc* classes. At 13:50:30, the browsing traffic rapidly

increased to the maximum capacity of the link; the *qdisc* for the synchronization traffic then falls to its allocated slice of 10 kbps, with no spare tokens to borrow from other classes. The browsing traffic then eases off at 13:53:00 and the synchronization service takes advantage of the spare capacity again, with the file synchronization completing at 13:57:00. From the plot it is clear that the traffic shaping for the synchronization service is very responsive to increases and decreases in capacity and makes full use of capacity when available but backs off to its ‘trickle’ rate when the link is saturated.

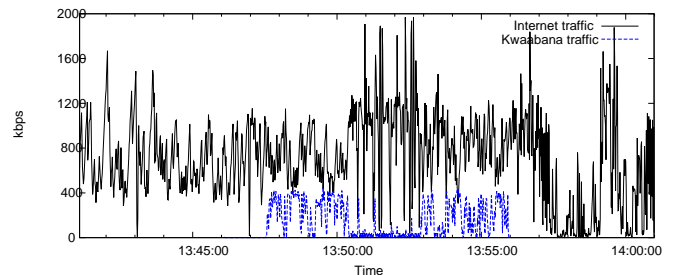


Figure 10: Plot of traffic passing through the gateway illustrating Kwaabana file synchronization and all other Internet traffic.

6. RELATED WORK

In our previous work, we investigated the lack of content generation in a remote community as well as the obstacles to uploading and sharing content [7, 8]. We believe that the facilitation of content sharing is tightly coupled to content generation; users will be encouraged to generate more content when this content can be successfully shared with local and international users.

Locality of online interactions was first observed with the advent of online social networks (OSNs). Wittie et al. [19] exploit locality of interest in order to improve OSN usability for remote areas. Their work is geared towards larger geographic regions (e.g. whole countries). Onnela et al. [11] analysed 72 million calls and 17 million text messages in Europe and found that probability of communication decreases by 5 orders of magnitude when distance between communicating parties increases from 1 km to 1000 km.

To cope with limited bandwidth, we propose a technique that only uses bandwidth capacity when it is available at any time of the day. Other techniques have been explored, such as time-delayed proxies for bandwidth-limited networks [4]. The value of local clouds in developing regions was explored in [5]; an order of magnitude improvement in performance was demonstrated with an application mirror installed in a local network. Collaborative caching was proposed by Isaacman and Martonosi [6], where cached content on client devices is made available to all clients in the local network. This solution acknowledges the value of local stored content as well as the importance of distributed access in unreliable networks. A system similar to the one proposed was implemented in [18]. These solutions, however, do not leverage the power of a local social network in sharing content locally, nor are they geared towards facilitating content generation.

7. DISCUSSION AND CONCLUSION

Deploying localized web-based applications in developing regions comes with a unique set of challenges. In many developing countries, Internet usage behaviour [7] and local traditions and culture [16] are different than in developed countries. Deployment requires patience and a series of refinement cycles in order to adapt the system to needs of the local community. Technical solutions designed according to western standards typically fall into rapid disuse.

We have found that a combination of qualitative analysis, using in-person or on-line surveys and community workshops, and quantitative analysis, using Internet traffic analysis, is very beneficial when designing web-based applications for developing regions such as Macha, Zambia. The ultimate purpose of the system may also evolve as users begin to use the system. For example, we were focussed on providing a file sharing service with Kwaabana, but discovered many users also use the system to backup their files. What may seem like an obvious choice of system when designing the initial solution – in our case, a Facebook-based file sharing solutions – has most likely overlooked a key constraint in the design, such as Internet costs. Rural users, especially experienced users like those in Macha, Zambia who have had Internet access for over 5 years, are used to high quality GUIs from cloud-based services on the Internet. Apart from the advantage of avoiding Internet access costs, user uptake increased substantially once we deployed a system based on ownCloud, which comes with a high-quality GUI.

As a local village network scales, it may be necessary to exploit locality of interest at a smaller scale than the current village-level granularity. For example, a cluster of users around a hospital or school may stream a large amount of content between each other. To exploit this scenario, we believe that the storage system used by Kwaabana could be distributed throughout the local network. A logical mechanism to do this would be to use a P2P storage solution, in which the files are distributed throughout the network on user computers. The P2P storage system could learn local sharing patterns using a social graph and attempt to store files on machines as close as possible to the users that are most likely to want to retrieve this content.

As the bandwidth divide increases between developing regions connected over slow Internet and the rest of the international community, services such as Kwaabana will be key to enabling vibrant collaboration amongst local users and sharing of locally generated content with the international community. Without these services, locally generated content will be trapped in digital storage islands in developing regions across the world.

8. ACKNOWLEDGEMENTS

This work was supported in part by NSF Network Science and Engineering award CNS-1064821. The authors would also like to thank Linknet in Macha, Zambia for their continued cooperation on this project.

9. REFERENCES

- [1] E. Brewer, M. Demmer, M. Ho, R. J. Honicky, J. Pal, M. Plauche, and S. Surana. The challenges of technology research for developing regions. *IEEE Computer*, 38(6):15–23, June 2005.
- [2] I. Buskens and S. Earl. Research for change outcome mapping’s contribution to emancipatory action research in africa. *Action Research*, 6(2):171–192, 2008.
- [3] J. Domčech, A. Pont, J. Sahuquillo, and J. A. Gil. A user-focused evaluation of web prefetching algorithms. *Computer communications*, 30(10):2213–2224, 2007.
- [4] B. Du, M. Demmer, and E. Brewer. Analysis of WWW traffic in Cambodia and Ghana. In *WWW’06*, Edinburgh, UK, May 2006.
- [5] A. Fahad, Z. Chen, K. Shen, J. Bigham, and A. Fahad. An evaluation of web acceleration techniques for the developing world. In *NSDR’12*, Boston, MA, June 2012.
- [6] S. Isaacman and M. Martonosi. Low-infrastructure methods to improve internet access for mobile users in emerging regions. In *WWW’11*, Hyderabad, India, March 2011.
- [7] D. Johnson, V. Pejovic, E. Belding, and G. van Stam. Traffic Characterization and Internet Usage in Rural Africa. In *WWW’11*, Hyderabad, India, March 2011.
- [8] D. Johnson, V. Pejovic, E. Belding, and G. van Stam. VillageShare: Facilitating Content Generation and Sharing in Rural Networks. In *ACM DEV*, Atlanta, GA, March 2012.
- [9] D. L. Johnson, E. Belding, and G. van Stam. Network Traffic Locality in a Rural African Village. In *ICTD’12*, Atlanta, GA, March 2012.
- [10] K. W. Matthee, G. Mweemba, A. V. Pais, G. van Stam, and M. Rijken. Bringing Internet connectivity to rural Zambia using a collaborative approach. In *ICTD’07*, Bangalore, India, Dec. 2007.
- [11] J. Onnela and S. Arbesman. Geographic constraints on social network groups. *PLoS one*, 6(4), 2011.
- [12] Owncloud: Cloud storage for private servers. <http://http://owncloud.org/>.
- [13] V. Pejovic, D. Johnson, M. Zheleva, E. Belding, L. Parks, and G. van Stam. The bandwidth divide: Obstacles to efficient broadband adoption in rural sub-saharan africa. *International Journal of Communication*, 6:2467–2491, 2012.
- [14] J. Postill. Localizing the internet beyond communities and networks. *New Media & Society*, 10(3):413, 2008.
- [15] C. Putnam and B. Kolko. The social meaning of icts: patterns of technology adoption and usage in context. In *ICTD’10*, London, December 2010.
- [16] G. van Stam. Observations from rural africa: An engineer involved in icts and critical ethnography in macha, zambia. University Lecture, 2012.
- [17] G. van Stam, D. Johnson, V. Pejovic, C. Mudenda, A. Sinzala, and D. van Greunen. Constraints for information and communications technologies implementation in rural zambia. In *Africomm’12*, Yaounde, Cameroon, November 2012.
- [18] W. W. Vithanage and A. S. Atukorale. Bassa: a time shifted web caching system for developing regions. In *NSDR’11*, Washington DC, June 2011.
- [19] M. Wittie, V. Pejovic, L. Deek, K. Almeroth, and B. Zhao. Exploiting Locality of Interest in Online Social Network. In *CoNEXT’10*, Philadelphia, PA, December 2010.