Name of organization supported	**University of Moratuwa**
Project title	**e-Mail system for Telecentres and Schools**
Dates covered by this report	**15/Mar/2009 - 30/Jun/2010**
Country where the project has been implemented	**Sri Lanka**
Project leader name	**Prof. Gihan Dias**
Team members (list)	**Dr. Sanath Jayasena - Project Co-Director, Mr. Chamara Dissanayake, Mr. Roshan Perera, Mr. Gayan Keppetipola, Mr. Sumedha Jayasinghe, Ms. Dinusha Jayawardene, Mr. Charith Mahawatta, Mr. Rohana Dasanayake, Ms. Mahiya Rafeek**
Submission date	**09/July/2010**

Table of Contents

Synthesis

Although technologies such as ADSL and 3G have brought broadband within the reach of ordinary people, many schools, telecenters and other small organizations still use kilobit-speed links. These connections are often priced based on data volume. This causes users of these sites two major problems:

1. Severe congestion on the link to the Internet resulting in poor performance; and
2. High cost due to volume of data transferred.

Email is still the most important application on the Internet. Many users use email daily, more often than other applications, and spend a considerable amount of time reading email each day.

Any delays encountered when accessing email will multiply, and may add up to a significant amount of time. For example, if a total delay of 60 seconds is encountered for each email message, and a user reads 20 messages a day, this adds up to over 8 hours of wasted time each month.

The objectives of this project are:

1. To provide users responsive access to email
2. To minimize the amount of email data transferred across the link from the Internet to the user's site.

It also allows users access to email even if the link is temporarily down. In addition, the system is being localised into our national languages, Sinhala and Tamil, in a separate project.

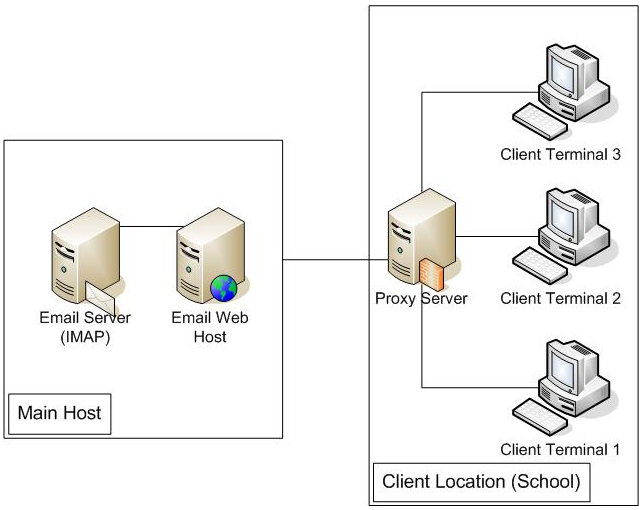


Figure 1: System Model

The system comprises a **central server** with good Internet connectivity. The initial central server is at the SchoolNet hub. Each remote site (school or telecenter) has its own **proxy server,** which provides a **webmail service** which is accessed using a web browser.

It is assumed that the link between the remote site and the central server has low bandwidth, and is often congested.

The following features were implemented to provide good performance and reduce the traffic on the link.

- **Prioritization and Fetching:** Emails are prioritized based on user preferences and are either pre-fetched (for high performance) or fetched on request (to reduce traffic on link).
- **Redundant Content Avoidance:** Content (such as attachments) are transferred only once.
- **Advanced Spam Filtering:** Spam is filtered at the central server, based on individual user selections.

The software and hardware output of the project comprises:

- The central server to receive email, process them, and forward them to remote sites.
- Proxy servers at each site to receive email from the central server, store them locally, and deliver them to the user on demand.
- Extending the Horde/Imp open-source web-based email system

During operational testing of the system, we observed an improved user response and a reduction in the volume of data.

The Project Team
L-R: Chamara, Roshan, Gayan (partly covered) Erandha, Sanath, Gihan, Sumedha, Dinusha, Kalana

Development Problem

Over a thousand Lankan schools have been connected to the Internet through SchoolNet. In addition, a large number of Nenasala telecenters have been established during the last few years, in addition to other 'Internet cafes'.

These provide the primary means of Internet access to many of our citizens.

These sites generally comprise up to 20 (often much less) computers that are connected to the Internet via a low-speed 128kb-512kb link. Due to cost and geographical constraints, it is generally not possible to significantly increase the speed of the link.

The main types of Internet usage are web, email, and telephony.

Although web access takes the larger share of both online time and data volume, email is still the most important application on the Internet. Many users use email daily, more often than other applications, and spend a considerable amount of time reading e-mail each day.

Any delays encountered when accessing email will multiply, and may add up to a significant amount of time. For example, if a total delay of 60 seconds is encountered for each email message, and a user reads 20 messages a day, this adds up to over 8 hours of wasted time each month.

The objectives of this project are:

1. To provide users **responsive** access to email
2. To minimize the amount of email **data transferred** across the link from the Internet to the user's site.

It also allows users access to email even if the link is temporarily down. In addition, the system is being localised into our national languages, Sinhala and Tamil, in a separate project.

Initially, we planned to customize an open-source email client, such as Mozilla Thunderbird. However, upon investigation we realized that school and telecenter users do not use such clients, but rely on webmail services such as Gmail. Therefore, we decided to use the open source Horde/Imp webmail server for this project.

We also identified that sites generally switch off equipment when closing the center in the evening. Therefore, we could not expect overnight downloading of files. Nevertheless, we decided to provide a server at each site, as the required performance could only be obtained by accessing a local server at each site.

Project Process

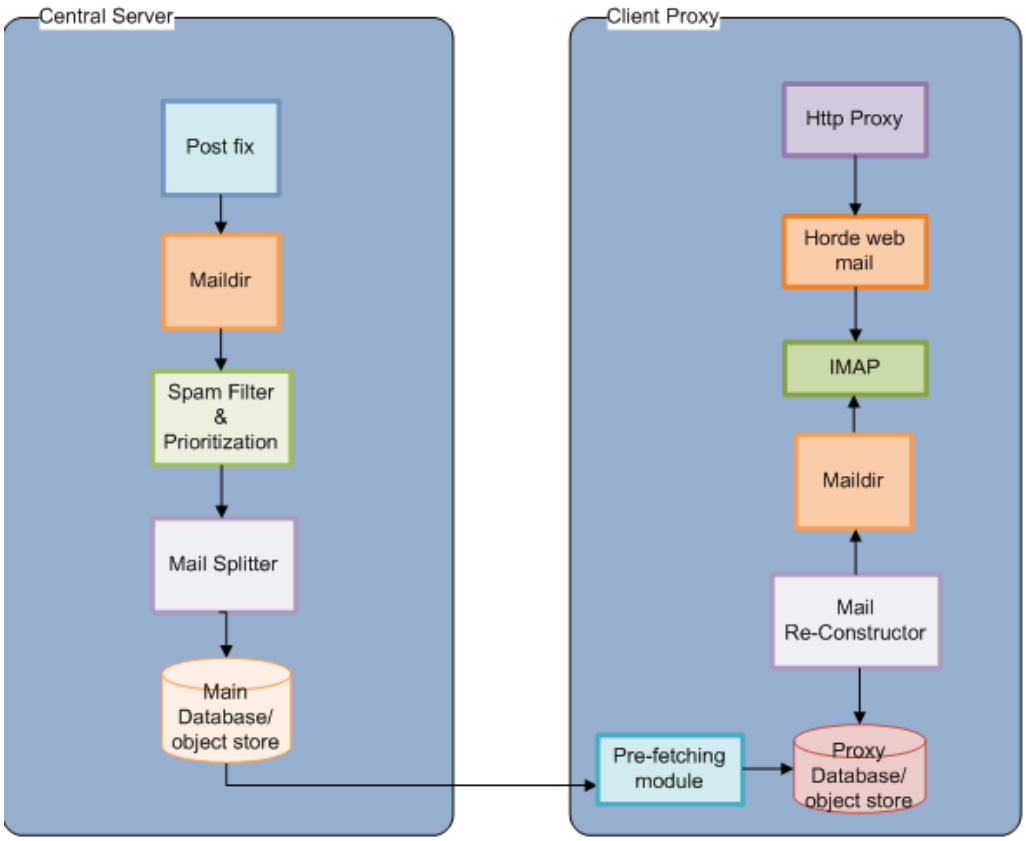


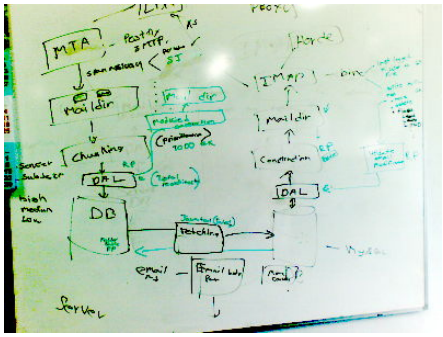
Figure 2: System Architecture

Processes successfully carried out

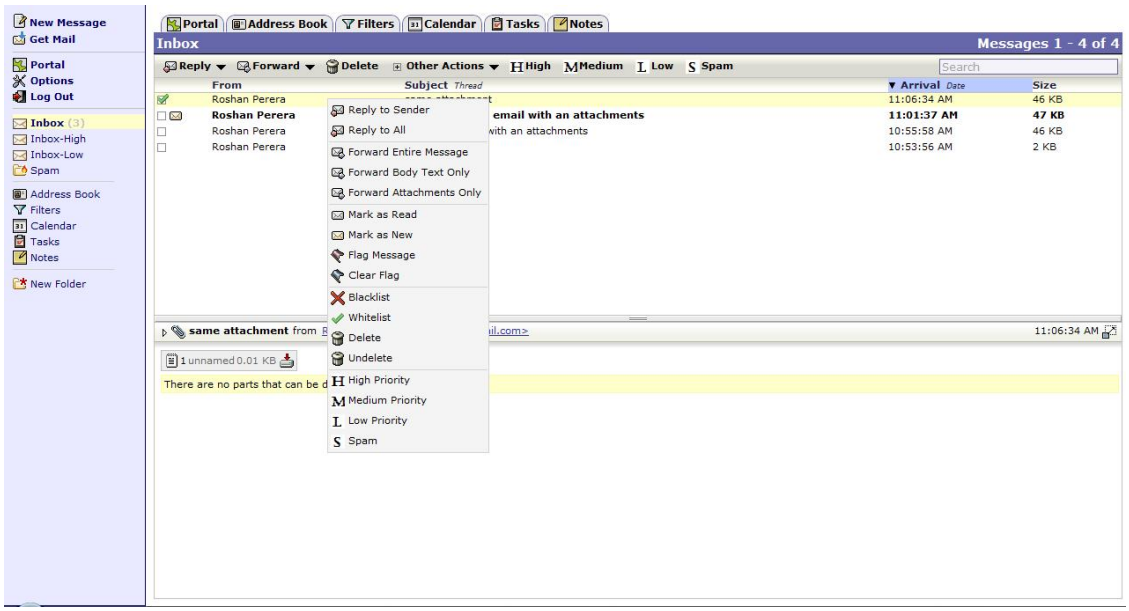
- Designing the system architecture: Over a series of meetings and email messages, the project team brainstormed the architecture and components. An innovative feature of our system is that it stores email in a database, rather than in a file system. The transfer of email between the central server and the proxy server is done by partial replication of the database between the central server and the proxy servers.

- Decomposition and composition of email messages: Email messages received at the central server are decomposed into components based on MIME type and stored in the database. Duplicated components (e.g. attachments) are de-duplicated and stored only once.

  When messages are requested by the IMAP server on the remote site, the proxy server composes the message by assembling the components.



- Customizing the email client: Horde was the selected email client for the project and was customized to allow the user to prioritize messages. The priorities are signalled to the central server via the IMAP server and the proxy server.



- Spam filtering: Spam filtering is based on SpamAssassin's Bayesean filter module running on the central server. The module was modified to re-train based on spam/non-spam categorization from each user, and to combine per-user and site-wide spam scoring.

Processes that did not work as intended

- End-to-end implementation of database: Our objective was to base the email system entirely on the database, and to dynamically generate e-mails at the remote site on demand. However, modifying the chosen IMAP server (BINC) to support this functionality proved difficult. Therefore, as an interim measure, we reconstructed the email files at each remote site, and the IMAP server accesses the files. We plan to upgrade the system to use the database end-to-end as a continuation of the project.

- Deployment at SchoolNet sites: Although Schoolnet informed us that they would make facilities available for testing and deployment, we were unable to obtain the facilities in time. Therefore, testing and pilot deployment has been carried out at the University of Moratuwa.

- User Training: Due to the unavailability of SchoolNet facilities, students could not be trained on the system. However, LAKapps carries out regular training on local language computing and computer usage (including SchoolNet e-mail), and we will incorporate user guides and training on this email system into future training programs.

Principal Findings
Our principal findings are:

1. Identification of the processes and constraints at schools and telecenters: Our proposal was based on the requirement of telecenter users, based on our experience. During the course of project development, we studied the processes and constraints at a number of schools and telecenters in several parts of the country, and identified several issues, including the use of webmail and the infeasibility of 24-hour operation of remote site proxy servers.

2. Evaluation and selection of suitable software for email system: Over 15 open-source, freeware and commercial products in the following categories were evaluated to identify suitable products to serve as the foundation of our system:

    - Email server
    - Spam filter
    - IMAP server and
    - Webmail system

3. Design and implementation of a database-based, central server / proxy server architecture email system: This may be identified as the central part of our system, and is the basis of our continuing work on the *eSriLankaMail* system. Basing an email system on a database is an innovative idea with very few public implementations, and we foresee a significant demand for such systems for large email servers. The central / proxy server email system architecture is also novel, and we expect it to be useful in other contexts, such as multi-branch corporate

email servers.

4. Redundant content management for email: This feature, though intended for reducing the volume of email traffic transferred, also reduces the storage requirements for email.

5. Prioritization, spam filtering, and transfer management: These features enabled the balancing of good user response times with avoidance of unnecessary data transfers.

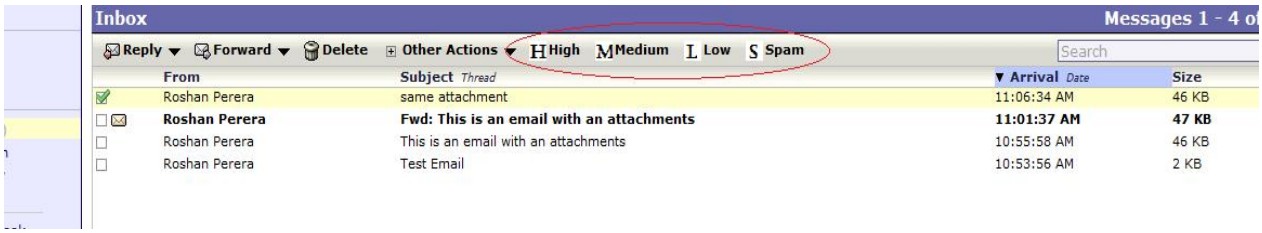
Fulfilment Of Objectives

Objective	***Progress***
Enhance an open-source web-based email software to improve performance on congested low-bandwidth networks	The system is completed and working. It comprises the following modules: 1. Spam Filter 2. Central Server 3. Remote sites Proxy Server 4. Data Transfer from central server to proxy server 5. Enhanced version of Horde webmail application
Host a localised server for the local community with enhanced functionality to help low-bandwidth clients connecting from various parts of the country	Done. The server is hosted at the LK Domain Registry, University of Moratuwa
Promotion and popularizing of the localised software	Four workshops were conducted in the Eastern province to provide awareness on computing in Tamil Language. Local (Tamil) language computing training is currently being conducted in Jaffna, Northern Province.
Deploying the client software at selected 5 pilot sites (“Nenasala” telecenters and schools)	The client software is deployed at the university and test sites.
Training 2x5=10 technical staff and 10x5=50 end-users at the 5 pilot sites on the localised versions of the software.	Four training programs were carried out at schools and universities.
Making the enhanced software localisable. (The localisation of the webmail software will be done outside this project).	Localizing the selected email client, Horde, to Sinhala and Tamil has been initiated as a separate project. Two teams are currently working on the translation.

Project Design and Implementation

Activity	***Period***	***Technologies implemented / methods and techniques used and any problems that arose***
Identification of software components	Initiation - 31 Dec 2008	IMAP - Binc server Email client - HORDE/IMP SPAM filter - spam assassin Proxy- Binc server - customized Mail - postfix
Database-based email system	Jan 2009 - Dec 2009	• The structure of the system and of the email messages were identified, and a set of database tables were defined to hold and process messages. • The mail transfer component was designed and implemented in Java. This component takes the message components, which need to be transferred to a given remote site, and synchronizes the central and proxy server databases. • Incoming messages are de-composed at the central server and stored in the database. • Initially, we planned to drive the IMAP server at the remote site directly from the proxy server database. However, as the integration of the selected IMAP server (Binc) with the database proved difficult, we finally decided to re-compose messages from the database into a file structure for access by the IMAP server.
Prioritization and Fetching module	Jul 2009 - Jun 2010	Emails are prioritized based on user preferences and are either pre-fetched (for high performance) or fetched on request (to reduce traffic on link). This module comprised the following components: 1. Enhancements to the Horde webmail system to allow users to identify priority and non-priority messages. 2. Identifications are signalled, via the IMAP server and database, to the prioritization module on the central server. The prioritization module updates its rule base based on the user selections, and transfers files appropriately.

Activity	***Period***	***Technologies implemented / methods and techniques used and any problems that arose***
Redundant Content Avoidance	Mar 2009 - Apr 2010	• Content (such as attachments) are transferred only once. • During the decomposition of incoming email, a checksum is computed for each MIME-part, and compared with the checksums of parts already in the database. If the checksum matches, the content is not stored redundantly, but the existing part is used. • This module was implemented in C++.
Advanced Spam Filtering	Mar 2009 - Jun 2010	• Spam is filtered at the central server, based on individual user selections. • We decided to use the Baysean Filtering method, as it has proved effective. Bayesean filtering is implemented in SpamAssassin, and initially we setup this feature. • During the second part of this development, we set up the system to train the Bayesean filter based on user categorization of messages, signalled by the prioritization module. • In the third part, the per-user filtering is integrated with the system-wide filtering.
Procurement of required hardware	June 2009 - June 2010	• Two PCs (one of which was initially used as the development server) were procured. • However, the supplier delayed the procurement of the system server. We are currently running the server on a borrowed computer.
Project issue tracking and management	Throughout the project	• Appointment of a project manager to identify issues and follow-up with developers. • Using a project issue log to track and monitor the issues. • Using SVN for software and documentation repository • Regular weekly project meetings were held, and meeting minutes prepared and circulated.



Mail Prioritization User Interface

Partnerships with other organizations, researchers, and community leaders

The project was carried out by the Centre for Localised Systems and Applications, University of Moratuwa in close collaboration with the Dept. of Computer Science and Engineering, and the LK Domain Registry.

The project benefited immensely from the localisation work carried out by the Centre, and the awareness and dissemination work carried out by the Centre and LK Domain Registry.

Our partnership with SchoolNet assisted us in obtaining access to schools and Zonal Computer Resource Centres (CRCs).

We have also built up a close relationship with the ICT Agency of Sri Lanka, which will assist in using this system at telecenters.

Disciplinary Orientation

This project is strongly technically oriented. However, our existing capacity in awareness, training and support helped us to bring the system to users.

Best Practices

One major issue we encountered is the low management capacity at many school labs and telecenters. Many PCs are non-operational, and repairs take a long time.

Therefore, the introduction of servers at remote sites is fraught with difficulty. It is anticipated that servers will break down frequently, and need to be repaired.

Project Outputs and Dissemination

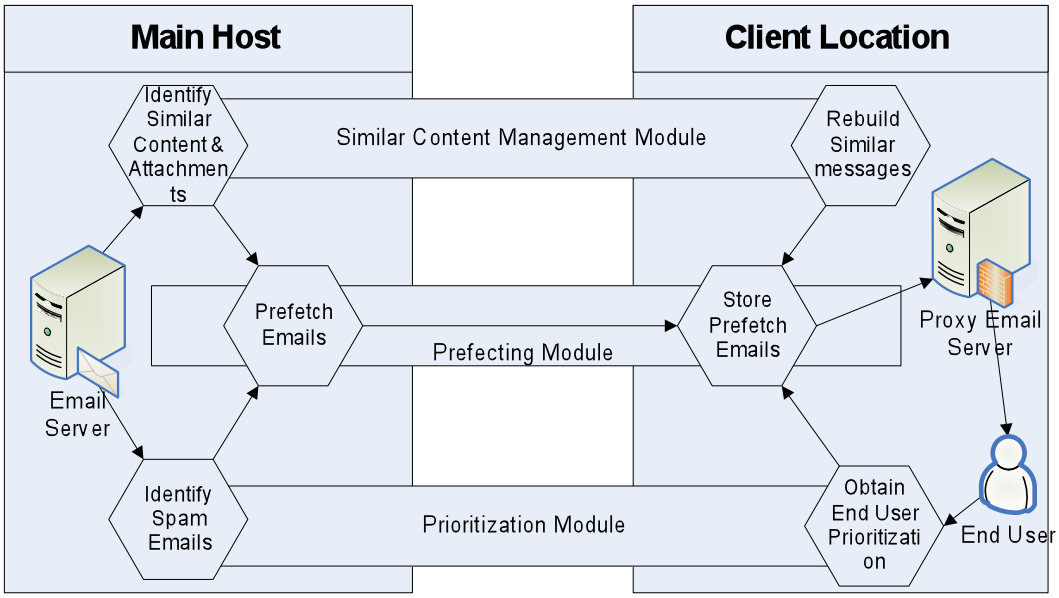
Project outputs comprise principally of open-source software.

The following applications were completed and are available for download from our site.

1. Central Server
2. Proxy Server
3. Webmail Service

The following modules are incorporated into the above applications.

- Prioritization and Fetching module
- Redundant Content Avoidance module



- Advanced Spam Filtering module

The central server system is installed on an operational server, and is available for use by remote sites. Additional outputs are the training and awareness programs being carried out.

Capacity building

Although the LAKapps center has previously carried out several projects which main focus was localisation, awareness, installation, support and/or training, this was the first software development project carried out.

In addition to the work carried out by part time staff, two full time software developers (who changed from time to time) were recruited for the project.

The availability of a full-time project manager was of major significance to the project.

Two PCs and a server were procured as a result of this project.

Project Management

The project was carried out under the administrative structure of the university, which follows procedures laid down by the government. Although sometimes restrictive, they allow us to comply with both local as well as donor requirements. The university finance division manages all funds in accordance with government procedures.

The project director managed the project, with the assistance of the project manager and administrative staff. The project was well managed, although delays were experienced.

One major problem was the high turnover of project staff. This not only resulted in delay, but also attrition of technical knowledge. Modification of modules written by staff members who have left the project proved difficult.

However, the morale and motivation of the current staff is high.

One major factor in project sustainability is the use of the open-source model. Wherever feasible, our modifications are upstreamed to each open source product, or alternatively made publicly available from our server.

Adoption of the system by SchoolNet will help ensure that bug fixes and updates are carried out.

The major factor, which would ensure sustainability, is the adoption of the system by the e-SriLankaMail project, which will expand on the current system and make it available nationally.

Impact

It is still too early to assess the impact of the system on target groups. An impact assessment will be carried out once a significant user base is built up.

Overall Assessment

The project has led to three principal innovations:

1. The development of a database-based email system, which allows the de-duplication of redundant content.

2. The development of a central / proxy-based system, which permits control of email content transfer and synchronization.

3. Identification of email priority and spam by individual users at remote sites, which is signalled to the server for future classification.

In our opinion, the project has been successful in achieving its objectives, relative to the investment of time, effort, and funding.

We believe that in the near future, the system will gain acceptance both in Sri Lanka and eventually in other countries.

The main weaknesses of the project currently are the lack of usage by external users. However the team is ready to accept bug reports and other observations from users as usage increases, and improve the software as needed.

The team is continuously working to improve the reliability of the software needs and the modules in need of improvement.

Recommendations

We are happy with the awarding, administration, and funding of the ISIF. The initial workshop in Hyderabad gave us an idea of the ISIF program administrative requirements, and put a face to peoples' names involved in ISIF. It also provided an opportunity to meet other practitioners in the same area.

Although in theory, the wiki and other online interaction systems should be useful, in practice we found them of limited use. We recommend to ISIF to adjust the online collaboration strategy, and clarify what in expected from the grant recipient in each case. We also recommend that the template for technical reports is reviewed and adjusted, to avoid repetitive information. We also recommend that the requirements in the report template are better described, so technical oriented team like ours can we work through the sections of the report focusing on social development and impact.

We found it difficult to ramp-up capacity and complete the project in one year, even with the extension granted. We suggest that projects have the option of achieving defined outputs in one year, and then obtaining follow-on funding for additional components for two additional years.