Demonstrating SWiFiIC - Sustainable WiFi for Rural Communication

Abhishek Thakur, Aarthi J, Amit Chawla, Chittaranjan Hota
Computer Science and Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
{abhishek, h2014103018, f2012412, hota}@hyderabad.bits-pilani.ac.in

Abstract — **Delay tolerant networks have not been operated for terrestrial communication on a long term basis. SWiFiIC is an effort to create an extensible platform for future applications in rural communication where delay and loss are less important as compared to cost.**

While costs are minimized, SWiFiIC targets financial sustainability by including a revenue model. As demands for infrastructure increases in rural communication with Internet of things and related applications, SWiFiIC attempts to provide a simple extensible middle-ware which is cost efficient to deploy and operate.

Keywords— Delay Tolerant Networks, Rural Communication, Wireless

I. Introduction

Delay Tolerant Networks [1] was proposed more than a decade ago with primary applications in deep space communication and other challenged environments like battlefield, underwater communication, deserted locations, post disaster communication etc. Routing and message transfer schemes for DTN are based on store-carry-forward mechanism.

In DTN, there is no end-to-end connectivity between the source and destination nodes. Instead the bundles (DTN messages) are transferred (relayed) from source nodes to other nodes when they have contact. These other nodes (also called intermediate nodes) act as custodians for the bundles. They receive a bundle from a node, physically carry the bundle from one place to another, and pass the bundle to some other node they meet. The latency of a bundle in DTN is unpredictable; it can be of the order of minutes, hours or even days depending on the deployment scenario.

For deep space communications or interplanetary networks, satellites act as custodians of the bundles. DTN has also been attempted for communication with rural communities and remote areas of developed countries [2] or semi-rural areas of developing countries [3]. In rural or remote scenarios people or vehicles moving through the area carry devices that participate in DTN. In such perspective these networks may also be termed as pocket switch networks (with people as carriers) or vehicular ad-hoc networks (with vehicles as custodians).

The author's team, in their prior work have analyzed [4] rural deployments of developing countries. It was noticed that most of the deployment have not sustained themselves without significant external funding. The primary reasons were found to be the complexity of solution and inability of the local community to generate revenue from such solutions.

To mitigate these issues the authors have designed a platform called SWiFiIC (Sustainable Wi-Fi in Indian Context) with focus on **rural** communication in developing countries like India. Over the last three years, the implementation of the design has progressed to a demonstrable proof of concept using open source DTN stack from Braunschweig University of Technology [6].

This paper is a demonstration of SWiFiIC as it exists today and has most of the implementation as proposed in [5]. Extension from [5] is related to change of some notations, usage of screen shots from running applications and discussions on challenges because of central Hub.

Structure of rest of the document is as follows: Section II provides a brief overview of the architecture components. Section III provides details of user interface for management applications. Section IV includes demonstration of a sample application on SWiFiIC. Section V includes details of challenges being observed during SWiFiIC development and deployment. Section VI includes conclusion and a broad overview of future work targeted by the team of authors.

II. Architecture

SWiFiIC platform targets four primary goals

- Simplicity of deployment and easy management.
- Usage of "off the shelf components" to benefit from commoditization to achieve cost efficiency and minimize operational cost.
- Commercial model to drive revenue generation.
- Extensible platform with ability to add applications.

To achieve these goals, a deployment of SWiFiIC uses a Linux based hub to control the deployment. All end user devices are Wi-Fi enabled smart phones or tablets. All the communication uses Wi-Fi (either as Wi-Fi direct or assisted by consumer grade access points).

The platform allows specific applications to have gateway / proxy capabilities extended using the Hub. E.g. broadband connectivity (like NOFN) to the village can be integrated to the Hub. In absence of wired or optical connectivity, 3G/4G can also be used for connectivity. If the cost of communication is a concern, these gateway elements can compress the information. Moreover the platform allows for integration of DTN as a transport for open source frameworks like IoTivity[7].

In this demonstration, the hub is not being used as a gateway for providing Internet based services. It only show-cases the communication within the village.

A. Physical Components

Hub: A low end laptop running Linux (2GB RAM, 250 GB hard disk, dual core). Linux has been chosen primarily to aid debugging during the initial prototyping phase. If needed, it can be moved to an Android device in future. The cost of the device was USD 200.

The software installed on the Hub is Ubuntu 14.04, IBR-DTN, MySQL, Tomcat, PHP and Apache services along with SWiFiIC components.

User terminals: End users have access to SWiFiIC platform using Android devices. All communication relies on Wi-Fi and happens using IBR-DTN service running on these devices. In addition a lightweight SWiFiIC User Terminal Application (SUTA) runs on these devices so that all SWiFiIC applications, list of users and basic billing information etc. can be made available to end users. Extensible applications make use of data provided by SUTA to further implement their functionality. E.g. Messenger gets the list of users.

Operator Terminal: With the intent that Hub need not be physically exposed, an Android device connects to services running on the Hub to provide all management and troubleshooting activities. E.g. User addition and modification, account recharge etc. are provided by the SWiFiIC Operator App (SOA) running on the Operator Terminal. SOA does not rely on DTN, instead it directly communicates with the Hub over HTTP.

B. Logical Components

There are platform specific components called SUTA and SOA. Other than these, SWiFiIC allows third party applications to be developed and deployed on the platform. Each independent App has two components – 1) Android application and 2) Hub specific code. In SWiFiIC paradigm the third party app is called Swilet. The hub component of Swilet is called Hublet, while the corresponding mobile applications are referred to as Moblets.

SUTA Moblet running on end-user device, provides access to the Swilet. As a new Swilet is deployed, the Hublet component automatically runs on the Hub. The details of the new Swilet are pushed (multicast) to all SUTA instances. Once the end user is notified, the end user may decide to install the Moblet piece through SUTA. This functionality is still under development.

Periodically, the SUTA Hublet multicasts the list of users to all SUTA Moblet instances. The SUTA Moblet also responds periodically to these heartbeat messages to provide feedback on round trip delay, node reachability etc. The response includes statistics of device like battery power, software version etc. This information is used by Hublet to update the list of active users and their device details.

The Moblet for other apps can query SUTA for SWiFiIC deployment specific information. Presently this is limited to details of Hub and the list of end-users in the SWiFiIC platform.

C. SWiFiIC communication

Except the SOA, all communication in SWiFiIC relies on unicast or multicast based Delay Tolerant Network. In the present iteration all Moblets talk to their respective Hublets (unicast). The Hublet code may relay or reply in unicast or multicast manner.

As mobile devices come in contact with each other, they exchange the messages so that they reach the destination. Because of reliance on DTN, many DTN bundles may not reach the destination or have significant delay (order of hours). The Swilets need to implement their own logic (like sequencing, acknowledgement, retry etc.) to ensure that it gets the level of quality they desire. Occasionally the end user may need to be notified, especially for people who live on the periphery of the village.

D. Cost and benefit of Centralized Hub

All messages are relayed to the Hub and then distributed back towards the recipients. This approach was necessary to ensure monitoring (and hence billing) of the setup. In future this will also help scan the messages to include IDS/IPS functionality. Further computationally heavy tasks or large storage of data can be delegated to Hublet, thus reducing the space and power needs on mobile devices. Hublets can also act as gateway/proxy to Internet.

Because of such a centralized communication, round trip delays will increase and delivery ratios may drop. In separate ongoing work, authors have done simulations to measure such impact. It was found that in the vicinity of the Hub, congestion sets in as load is increased. To mitigate this, in future design, the team intends to have multiple hubs directly connected over wire.

One benefit that the centralized design allows is the ability to measure quality of service for end users. If the SWiFiIC operator finds that some devices are having poor connectivity, she/he may decide to use dedicated ferry based mechanism to serve such nodes. For example they may hire a person to periodically visit the area where the poorly served nodes reside, or they may allow higher replicas to be created for messages originating or terminating at such nodes.

III. BASE PLATFORM COMPONENTS

SOA and SUTA are two crucial parts of the SWiFiIC platform. Moreover libraries are provided for Hublet and Moblet pieces to simplify development. Figure 1 below captures the deployment for Hub. Figure 2 captures the deployment for typical end user terminal.

A. *Hublet basic Logic*

For each Swilet, the Hublet needs to implement the handlePayload function. This is the code that gets triggered on receipt of message from Moblets.

Algorithm 1: Handle Payload

function handlePayload(payload, context)
  Identify the *operation* within the payload
  Check account previleges of the originator of payload
  Do the necessary billing for the *operation*
  Invoke the implementation for *operation*
End

handlePayload is the single entry to process all types of operations for the specific Swilet. E.g. for Exam Swilet, it handles operations like, addStudent, sendPaper, submitAnswerScript, getAnswerScripts, getAttendance etc.

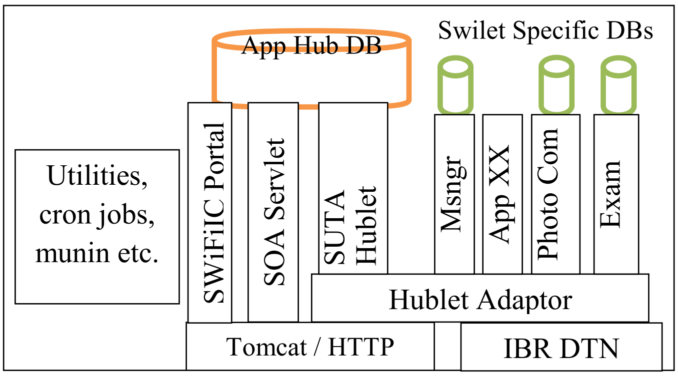


Figure 1 Deployment on Hub

A check on account privilege is necessary to verify the user and avoid impersonation. Billing aspects are optional. The implementation of operation is Swilet specific logic and generally operations such as queries to Swilet specific DBs and generation of DTN messages back towards Moblets.

Hublet code relies on functionality provided by generic code within the Hublet Adaptor so as to mask the complexity of communication.

B. *Moblet basic Logic*

The Moblet's in present context are Android applications. Other than implementing the logic for the user interaction and application specific requirements, the Moblet's implement a broadcast receiver to handle incoming messages from the Hublet. Further they rely on base libraries (SWiFIC JAR) to register with the IBR DTN service and to send / receive their messages.

For getting the context of SWiFIC deployment like Hub's DTN address, list of users etc., Moblets use content provider interface of SUTA.

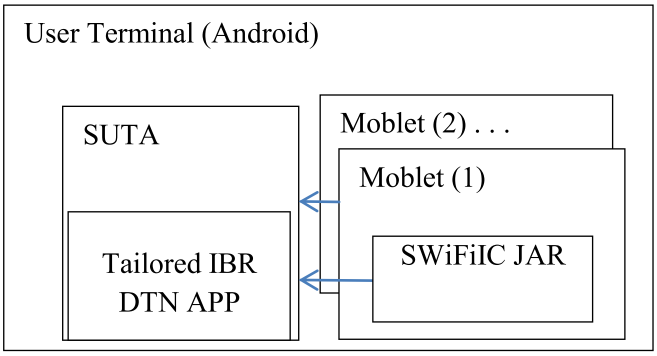


Figure 2 Deployments on User Devices

C. *Screen shots*

Following figures captures the screen shots for different UI components.

1) *SOA*

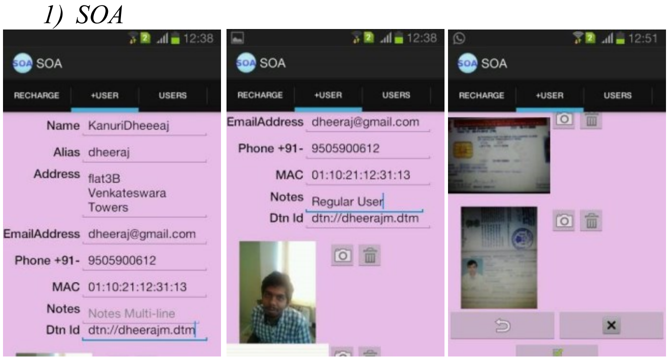


Figure 3 SOA – Add user

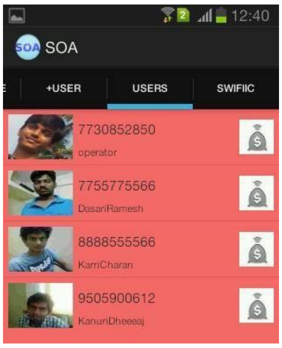


Figure 4 SOA – Other Screens

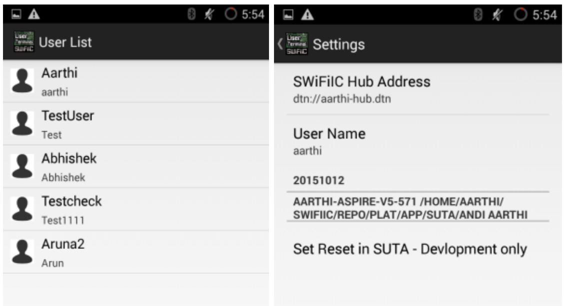


Figure 5 SUTA Moblet

The SUTA Hublet code receives the updates from the Moblets and also generates periodic updates. The figure below captured debug output for some of these scenarios.

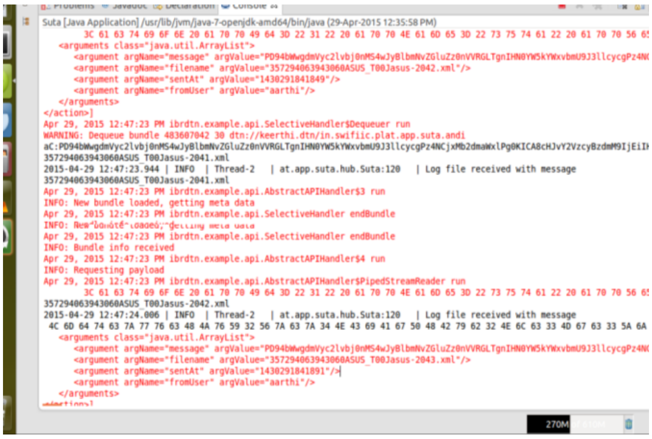


Figure 6 SUTA – Hublet – Node Updates Received

IV. Sample Application

Few applications have been developed to demonstrate the capabilities for the platform. We discuss the messenger app implementation below including a few screen captures.

A. *Messenger*

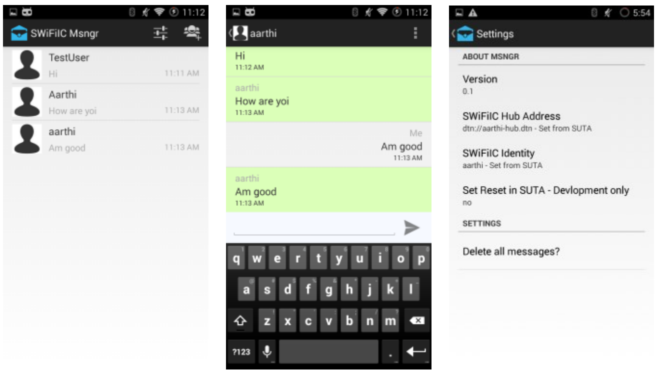


Figure 7 Messenger App – Multiple screens

Hublet: The messenger Hublet after authenticating the sender, debits (notional amount of 1 rupee presently) from their account. Thereafter it queries the DTN id for the receiver of the message and relays the message to the destination DTN node.

Moblet: Majority of the code here is related to UI and tracking of messages received in the past. In addition using the broadcast receiver the code handles incoming DTN messages. To send a new message (towards Hublet) the code uses sendAction on Helper functionality provided by the SWiFiIC JAR files.

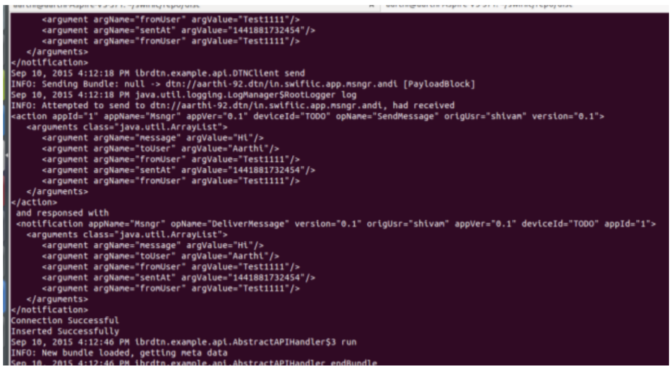
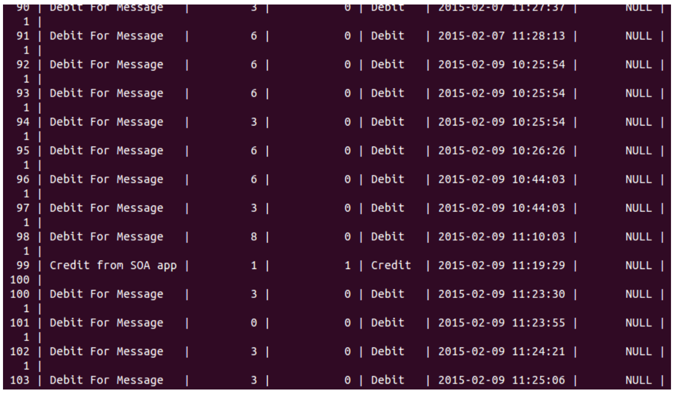


Figure 8 Messenger Hublet – Logs



```
 91 | Debit For Message |  6 |  0 | Debit  | 2015-02-07 11:28:13 | NULL |
 92 | Debit For Message |  6 |  0 | Debit  | 2015-02-09 10:25:54 | NULL |
 93 | Debit For Message |  6 |  0 | Debit  | 2015-02-09 10:25:54 | NULL |
 94 | Debit For Message |  3 |  0 | Debit  | 2015-02-09 10:25:54 | NULL |
 95 | Debit For Message |  6 |  0 | Debit  | 2015-02-09 10:26:26 | NULL |
 96 | Debit For Message |  6 |  0 | Debit  | 2015-02-09 10:44:03 | NULL |
 97 | Debit For Message |  3 |  0 | Debit  | 2015-02-09 10:44:03 | NULL |
 98 | Debit For Message |  8 |  0 | Debit  | 2015-02-09 11:10:03 | NULL |
 99 | Credit from SOA app | 1 | 1 | Credit | 2015-02-09 11:19:29 | NULL |
100 | Debit For Message |  3 |  0 | Debit  | 2015-02-09 11:23:30 | NULL |
101 | Debit For Message |  0 |  0 | Debit  | 2015-02-09 11:23:55 | NULL |
102 | Debit For Message |  3 |  0 | Debit  | 2015-02-09 11:24:21 | NULL |
103 | Debit For Message |  3 |  0 | Debit  | 2015-02-09 11:25:06 | NULL |
```

Figure 9 Hub – User transaction ledger entries

V. Development and Deployment Challenges

During the present development and trials for SWiFiIC, it has been noticed that user experience for Wi-Fi Direct is sub optimal. It confuses some of the end users who do not understand the technology in depth.

The peer discovery for IBR-DTN relies on multicast support. In many scenarios for Android devices, the peers have not been discovered even after few minutes. Restarting the Wi-Fi interface or restarting the IBR-DTN service frequently solves this issue. It was also observed that some access points allow peer discovery to happen most of the time, while a few do not.

The author’s team has also surveyed a nearby village to better understand the challenges for deployment. While some villagers can afford their own devices, majority of the others still cannot afford to spend four thousand rupees to have their own tablet. It is an unsolved challenge to create a viable model where the devices can be provided on a subsidized basis, without the fear of loss/theft.

VI. Conclusion and Future Work

SWiFiIC platform is being operated by multiple students within the campus of the authors. It is still far from production ready. Some of the pending items are

- Providing a better way to use Wi-Fi Direct.
- Improving diagnostics and reporting from SUTA.
- Developing other apps.
- Deployment in true rural scenario.
- Multiple Hub with wired connectivity so as to avoid single point of failure and reduce congestion.
- Auditing mechanisms to ensure that the Hublet does not go beyond its manifest.

The authors and the team of students continue to implement further functionality on an ongoing basis. The project code is maintained as open source at www.swifiic.in.

Acknowledgement

The authors are thankful to the development team of IBR-DTN [6]. SWiFiIC platform relies heavily on it. We also appreciate the feedback of reviewers who have helped improve the quality of this work.

References

[1] Fall, K. (2003, August). A delay-tolerant network architecture for challenged Internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 27-34). ACM.

[2] Grasic, Samo, and Anders Lindgren. "Revisiting a remote village scenario and its DTN routing objective." Computer Communications 48 (2014): 133-140.

[3] Pentland, Alex Sandy, Richard Fletcher, and Amir Hasson. "Daknet: Rethinking connectivity in developing nations." Computer 37.1 (2004): 78-83.

[4] Thakur, A., & Hota, C. (2013, August). Sustainable wireless Internet connectivity for rural areas. In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on (pp. 1335-1340). IEEE.

[5] Thakur, Abhishek, and Chittaranjan Hota. "Designing an extensible communication platform for rural area." In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, pp. 1348-1355. IEEE, 2014.

[6] Schildt, Sebastian, et al. "IBRDTN: A lightweight, modular and highly portable Bundle Protocol implementation." Electronic Communications of the EASST. 2011.

[7] Linux Foundation Collaborative Project , “IOT Architecture Overview” https://www.iotivity.org/documentation/architecture-overview