

Simplifying Fault Diagnosis in Locally Managed Rural WiFi Networks

Sonesh Surana
University of California,
Berkeley
sonesh@cs.berkeley.edu

Rabin Patra
University of California,
Berkeley
rkpatra@cs.berkeley.edu

Eric Brewer
University of California,
Berkeley and
Intel Research, Berkeley
brewer@cs.berkeley.edu

ABSTRACT

The last three years have seen a lot of work in making WiFi-enabled Long Distance (WiLD) networking a reality in rural areas. Generally these networks are managed by non-local users who cannot guarantee long term support beyond a pilot. For long term operational sustainability, it is essential that maintenance duties be transferred to local administrators. In this paper, we argue that the research agenda should expand into areas of simplified diagnosis solutions as an enabler for locally managed WiLD networks. Motivated by real faults we have seen in our own deployment at the Aravind Eye Hospital, we propose a framework to simplify diagnosis and show some initial results towards this direction.

Categories and Subject Descriptors

C.2.3 [Network Operations]: Network Management, Network Monitoring

General Terms

Design, Management, Measurement, Reliability

Keywords

Fault Diagnosis, Rural WiFi, Operational Sustainability, Developing Regions

1. INTRODUCTION

Although there has been much active research on wireless networking technologies for developing regions, one important area tends to be overlooked: the ability of local staff to keep a system running over the long term. Software or hardware errors, power issues, or other transient environmental issues such as wireless interference can cause outages leading to poor experiences for users (and administrators). As these new systems strive for acceptance, outages can chase away potential users and jeopardize viability.

In two years of experience with rural wireless deployments, we have seen that an inability by local rural staff to fix errors

causes extensive outages and prevents the healthy growth and expansion of the network.

There are several specific reasons why maintenance in rural areas is hard. First, local staff tend to start with limited knowledge about wireless networking. This leads to limited diagnostic capabilities, inadvertent equipment misuse, and misconfiguration. Thus management tools need to help with diagnosis and must be educational in nature. Training helps as well, but high IT turnover limits the effectiveness, so education must be ongoing and part of the process.

Second, the chances of hardware failures are higher as a result of poor power quality. Although we have not conclusively inferred the failure rate of equipment for power reasons in rural areas, we have lost far more routers and adapters for power reasons in rural India than we have lost in our Bay Area testbed. This calls for a solution that provides stable and quality power to equipment in the field.

Third, many locations with wireless nodes, especially relays, are quite remote. It is important to avoid unnecessary visits to remote locations. Also, we should enable preventive maintenance during the visits that do occur. For example, gradual signal strength degradation could imply cable replacement or antenna realignment during a normal visit.

Fourth, the wireless deployment, although connecting local nodes, may not be accessible remotely or through the Internet. The failure of a single link might make parts of network unreachable although the nodes themselves might be functional. This makes it very hard for remote experts in another town or even local administrators to resolve or even diagnose the problem. This points to a need for a low-cost alternate or “back channel” (e.g. SMS) that allows remote access to the nodes even in the event of a failure of the primary link.

Overall, troubleshooting is hard even for experienced users or experts. The troubleshooting decision tree is not always obvious, which makes it harder to design guidelines for rural users. Users or administrators have to hunt for data (e.g. run ifconfig, ping reachability scripts, log into remote nodes) to isolate the fault. Although some this can be automated and even visualized, we may still not know the root cause of the fault. Some amount of non-trivial domain knowledge will still be needed to perform the actual diagnosis. And many times, even experts resort to hopeful reboots without a clear understanding of what is wrong.

We argue that these troubles exist in large part because the research community has not tried very hard: these systems are not designed or deployed with support for easy diagnosis built in right from the start. We propose a frame-

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.

NSDR'07, August 27, 2007, Kyoto, Japan.

Copyright 2007 ACM 978-1-59593-787-2/07/0008 ...\$5.00.

work for deploying systems with support for diagnosis even in the presence of primary faults, which we expect will improve the operational sustainability of the network. In addition to a data collection and monitoring infrastructure that can operate over intermittent networks or alternate channels, we suggest building hardware and software modules that can be queried independently from the primary ways in which they are normally accessible. This allows for diagnosis of subsystems that would not be available when the primary link is down. Note that this redundancy is mainly for diagnosis rather than failover recovery, which is typically much more expensive. Sometimes the redundancy can also be used for failure recovery as well as for diagnosis, which we exploit when possible.

We present our first cut at a framework for simplified diagnosis, and hope that it will influence how future deployments are planned and positively impact their operational sustainability. Specifically, we present our ideas on using cellphone back channels, network-addressable solar power controllers, virtual links to isolate link-layer and IP-layer diagnosis, and also our data collection architecture called PhoneHome. In Section 2 we provide some background of our wireless deployment and some lessons we have learned from it. In Section 3, we present some general principles that deployments should incorporate to improve operational sustainability. In Section 4, we present a set of architectural options, and then our initial results in Section 5.

2. BACKGROUND AND MOTIVATION

Recent work, including some of our own, in WiFi-enabled Long Distance Networking (WiLD) for rural connectivity has thus far focused on understanding the basic issues, characterizing performance, re-designing the 802.11 MAC, and planning network deployments [9, 10, 8, 3, 14, 7, 13, 12].

Over the last three years, many projects using WiLD connectivity have also been deployed for applications such as healthcare and education. A few examples are the Ashwini [1] project, Digital Gangetic Plains [6], CRCNet [4], and our own Aravind Eye Hospital telemedicine project [15].

Figure 1 shows the Aravind telemedicine network which connects the main eye hospital in Theni in southern India to five rural clinics, most of them through relays (to achieve line-of-sight). The WiLD links are primarily used for high quality videoconferencing (roughly 300-500 Kbps per stream) between rural patients and doctors at the main hospital for remote eye-care consultations. The links have enabled rural clinics in areas where there were no other options for eye-care. Between Jan 2006 and Feb 2007, the network has supported 18210 remote video-consultations.

Initially, we were involved in all aspects of planning, deployment, and maintenance of the network. Now local staff with the help of a local wireless vendor take care of network planning, link installation and some very basic maintenance. But we still provide bulk of the maintenance by diagnosing faults remotely and providing the local wireless vendor with instructions for fixing problems.

Theni is connected to its sister hospital in Madurai over satellite link and has Internet connectivity through a Madurai-based HTTP proxy. We use this link to collect data from the Theni network. However, of all the times we have tried to collect data remotely, the satellite link has been down 35% of the time. As a result, many instances of faults are brought to our attention by rural staff by email many

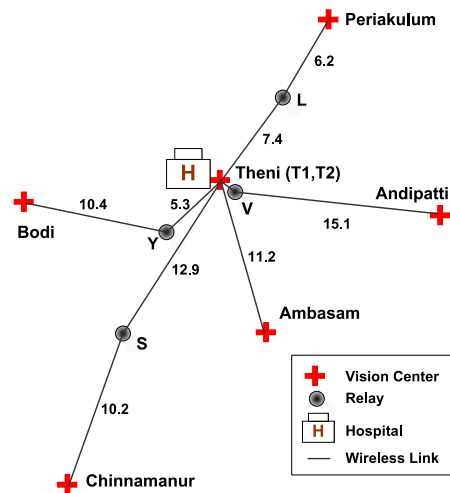


Figure 1: Aravind Telemedicine Network. Theni hospital is connected to 5 rural clinics. Theni has two wireless nodes, T1 and T2. Distances are in kilometers (Km).

days later when the link is back up, and sometimes the link is down for this entire period as they were not able to complete the diagnosis.

The most common description of a fault by our rural partners is that the “link is down”. This typically means that the videoconference is no longer working. Table 1 lists the range of faults that could result in a “link being down”, besides application-level errors. As seen in the table, there are a variety of reasons for link outages and it is not always easy to diagnose the root cause of the fault. For example a link may be down even if the remote node is up, but since the link is down, it is not possible to query the remote node for data to aid diagnosis. Accurate diagnosis can considerably save a lot of time and effort. There have been many instances where rural staff have gone to the remote site with difficulty only to realize it was a regular power shutdown from the grid or that it was a software problem which could have been fixed if there were an alternate low cost back channel to the router.

In the next section, we present our learnings on what a framework for diagnosis should include for rural networks. The central ideas are that we need data collection and monitoring working over intermittent links or low bandwidth back channels. We also need independent modules in the system that do not share fate with the primary link and can be queried even in the event of failure of the primary link.

3. REQUIREMENTS FOR DIAGNOSIS

Using examples of real fault situations from our experience with the Aravind network, we identify the key requirements and mechanisms that we need to implement in any real diagnosis system designed for rural WiFi networks.

3.1 Monitoring

Generally, it is difficult for remote experts to log in and administer these networks (e.g., the Aravind network is hidden behind a NAT and an HTTP proxy). Even local administrators find it hard to log in to individual routers to collect monitoring data because the complete network is not up all the time.

| Type | # | Fault description |
|------|-----|--|
| HW | 63 | Router board not powered on (grid outage, battery dead) |
| | > 7 | Router powered but wedged (low voltage, corrupt CF cards) |
| | 21 | Router powered but not connected to remote LAN (loose ethernet cables, burnt ethernet ports) |
| | 3 | Router on, but wireless cards not transmitting due to low supplied voltage |
| | 1 | Router on, but pigtailed not connected or other RF connectors gone bad |
| | 1 | Router on, but antenna misaligned |
| SW | 4 | No default gateway specified |
| | 3 | Wrong ESSID, channel, mode |
| | 2 | Wrong IP address |
| | 2 | Misconfigured routing |
| | | Driver errors, wireless cards not recognized |

Table 1: Various hardware and software errors that can result in link being down. The number of occurrences are an under-estimate since they are based on what we saw and not fully based on what the local staff experienced as well. This is because they did not keep accurate logs of the faults, partly because of the difficulty in diagnosis and tried various solutions till the fault was “fixed”.

3.1.1 Need to monitor status

Example: Network status: When the connection to an end node goes down, we want local admins to find out the extent of the problem and find out what parts of the network are unreachable as soon as possible. Without any monitoring, admins have to query all the nodes manually to find the point of failure in the network. They also have to wait for reports or phone calls from users at the end-points.

The admins need an infrastructure that continuously probes all the nodes and relays (e.g. ping tests) and presents the *current reachability* of the network in a graphical form at a central place. Finally, when nodes or relays go down, admins should be notified that part of the network is down for possibly proactive action.

3.1.2 Need to predict behavior

Example: Disk health: We have seen that often the data partitions of the compact flash disks are full and we cannot write any more logs to them. Due to frequent reboots, the `ext2` blocks become unusable. There are also hardware errors on the disk, and once they accumulate to a certain point, we have to replace the disk. If we do not periodically run `fsck` on the data partitions, they can become unusable, leading to services unexpectedly crashing. Thus we need to ensure constant monitoring of the file system health to *predict disk replacement* and to perform periodic repair.

Example: Predict uptime: We have to use battery backups on most of our relay points where there is no grid power. On loss of grid power, the battery might only last for some unknown amount of time after which the network will go down; typically rural staff has no idea when the battery will discharge. Therefore, we need the ability to monitor the remaining battery charge, and use that to *predict the immediate remaining uptime of the network* and schedule the opening or closing of the video conferencing.

Example: Predict battery lifetime: Also, battery life is often limited by the maximum number of deep cycle operations permitted. We need to *predict the expected lifetime* of batteries by keeping a count of deep discharges or by tracking how the rate of discharge changes over time (faster rates imply the battery is nearing the end of its life).

3.1.3 Need to compare expected behavior

Example: Signal strength: If a link is down but we can verify that both routers are up and functioning and that the wireless configuration is correct, then there might be a problem with the antenna. It is possible that the antenna is misaligned, or that some antenna cable or connector is disconnected.

However, if we can compare the measured signal strength of the target radio at the other local antennas before and after the problem, we can identify some of the causes. If the local antennas¹ see a lower RSSI than usual, the pigtail might be disconnected but if the local antennas still see very good RSSI, the antenna might be misaligned.

If the signal strength drifts to lower values over time, the antenna may be moving out of alignment. In general we need the ability to compare current behavior with generally expected behavior and derive conclusions from it.

3.2 Independent Diagnosis Modules

Although the best solution might be to have fully redundant systems, they are often too expensive. An intermediate solution is to have some independent hardware modules that enable diagnosis (but not full functionality).

3.2.1 Need to have back channels

It is hard to debug these networks because the only link to remote nodes are the wireless links themselves. As a result we need to build alternate mechanisms to reach remote nodes or query systems inaccessible by primary links.

Example: Network misconfiguration: Some common problems are simple network misconfigurations where IP addresses on the two ends are assigned differently. Although the link may show up as associated, it is not functional. A related problem occurs when there is a routing misconfiguration or loop; a particular node may be reachable but the reverse path may not work; thus pings fail to return.

Such problems can be diagnosed if we have an additional virtual link that uses a different IP subnet while still running on the same physical link. This virtual link configuration should be independent of any network configuration. Thus, a hop-by-hop login method would be acceptable for diagnosis.

Example: Independent channel: However if the problem is more serious than routing misconfiguration, we would not be able to access the remote node at all. It is impossible to distinguish between a power shutdown at the remote end, a board failure or a malfunctioning wireless card.

We need the ability to *access the remote nodes independently* from the primary wireless link. Ideally, this should be via a different wireless technology that has better propagation, but it can be low bandwidth, since it is only used for diagnosis.

¹typically we mount an extra omni antenna connected to a secondary radio at each node

3.2.2 Need for separate hardware control

Example: Hard reboots: A link might be down because the board might have reached a state where it needs to be rebooted. This could be because a wireless driver might have crashed or poor power might have caused the board to enter an unknown state. We have known cases where the wireless card would not transmit packets because of low voltage. In these situations where a simple reboot would solve the problem, we have had to physically go to the remote node just to power cycle it.

We need an *independent hardware based module* that reboots the system when it does not receive periodic heartbeats.

Example: Local connectivity: One common problem with the router boards is that the main ethernet port often stops working because of power spikes or lightning strikes on long exposed cables carrying power (PoE). This port is usually connected to PCs at vision centers. However these boards have secondary ethernet ports as well that can be used to diagnose (we have often done that to debug problems with routing and IP addresses misconfiguration). We need the ability to use extra ethernet or wireless ports for local diagnosis of problems.

3.2.3 Need for independent software services

Example: Software watchdog: We have seen problems where the routing daemon dies or goes into a bad state. This might result in nodes becoming unreachable. There are also situations in which the wireless driver goes into a bad state that stops the card from receiving or transmitting packets even while the OS still keeps running.

For all these situations, we need to have a *monitoring service* for essential software services on the router that can either restart the malfunctioning services or reboot the router.

Example: Upgrades: Remote upgrades to a router can often go wrong if the new kernel does not boot at all, or the wireless drivers do not load the cards correctly, or a network configuration change renders the router unreachable.

We need a *safe fallback mechanism*. In some cases we need the ability to set a *timeout period for new configurations* where we want to test it with the guarantee that the system would go back to a safe state after the specified timeout.

4. DIAGNOSIS ARCHITECTURE

In this section we describe the various architectural components required for the monitoring and diagnosis of a wireless network.

Figure 2 illustrates the components of a single wireless node. Each wireless node consists of the router board with several options for power. A power controller module consolidates all the power input options and provides regulated power to the board. The router board can have multiple antenna connectors for the point-to-point wireless links to neighboring nodes and a back channel component (cellphone). These components can talk to each other using ethernet or serial communication.

4.1 Monitoring Infrastructure

The objective is to provide status information that can be then used to either predict future failures or to diagnose problems using historical behavior. This infrastructure pro-

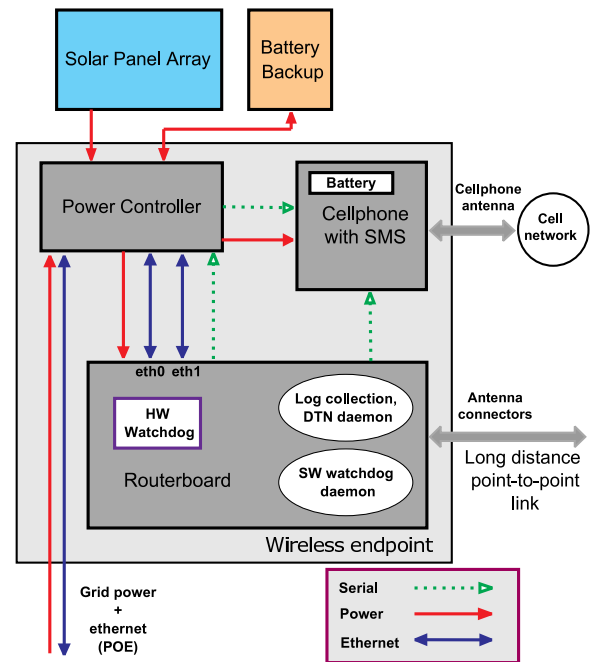


Figure 2: Components of a wireless node. Shows the basic router with power supplies and the additional components needed for remote diagnosis.

vides monitoring for local administrators by collecting data at the local central node of the network. It also enables remote monitoring by experts by tunneling data from behind NAT on low bandwidth connections.

Log collection: Each node in the network collects passive logs about network status information (IP addresses, interfaces), wireless status (channel, bitrate, association, received signal strength), the routing table (current set of dynamic routes, default routes), kernel messages, disk health status, etc. Each node also runs active tests to find out reachability to neighboring nodes and the quality of the link (latency, loss and throughput).

These logs are then posted regularly to the central node. To handle disconnections, we use a DTN [5] store and forward overlay network to transfers logs.

These logs can be used for: a) presenting the current reachability and status of the whole network, b) getting proactive messages to the admins about failures in the network, c) predicting future behavior of network or suggesting preventive measures, d) comparing with expected behavior to infer the possible causes of a problem, once a fault has occurred.

4.2 Wireless Node Components

The objective of the design of the wireless node architecture is to have components with independent failure models which can be used in the diagnosis of the network hardware and links even when the principle modules exhibit faults.

4.2.1 Independent Hardware Control

The independent hardware components are used to reboot the router board in case of anomalous state is detected. The objective is to use simple and robust components that are directly connected to either the router board or the power supply and can monitor it using a serial or ethernet port.

Option 1: Hardware watchdog: This is a form of independent control on the board itself which reboots the board if the OS fails to poke a particular register at a regular interval.

Option 2: Local power controller: The local power controller feeds power to the board from a combination of various input power methods: grid, solar panels or battery. It monitors the board using the serial port and reboots it whenever it does not receive a response from the board on the serial port. The power controller reports battery and solar panel status on the ethernet network. It can also implement failover capability using multiple ethernet ports of the router board.

4.2.2 Independent Back Channels

We would need to have an independent back channel to diagnose the network when the primary wireless link goes down. We can achieve this at different levels of abstraction.

Option 1: Local link addressing: The most basic mechanism to have independent channels is using Link Local IP addressing [11] where each link also gets local automatic IP addresses from a pre-assigned subnet that would work even when the system wide routing does not work. This can also be implemented by using virtual interfaces in the Atheros wireless driver [2].

Option 2: SMS on mobile network: This mechanism consists of an SMS channel over the cellphone network which can be queried in case of the failure of the primary wireless link. The SMS reply would have power parameters (grid power, remaining battery, voltage level of power supply), and basic status from the wireless board if it is up. This is often feasible because many places have basic cellphone connectivity (All our rural clinics have some degree of coverage provided by 2-3 providers at least).

For SMS as an alternate channel there are several options. A PCMCIA GSM or CDMA card can be used on the board. However, it would share fate with the router board. A regular cellphone can be placed inside the enclosure. It can use the serial port to query the board and the power controller and would be powered by the same source as the router. This is a simple modular design with a greater degree of independence. Even though the power source is the same, the cell phone can be on standby battery power for about a week and can still answer SMS queries about the system.

4.2.3 Independent Services

The objective is to have software mechanisms to restart failed services and provide support for safe upgrades.

Software watchdog: A software watchdog service would periodically use heartbeats to a set of services such as the routing daemon, DTN daemon (for logs) and restart them on detecting a crash. It would also query the network interfaces to make sure that the wireless cards are responding.

Safe upgrades: On an upgrade of the router OS, this service is configured to look for parameters which check that the upgrade does not violate any required properties. For example, the board should be able to initialize all the drivers, and ping local interfaces and remote nodes as well. If these are not satisfied, we should go back to a previously known safe OS state. This can be combined with the hardware watchdog mechanism in conjunction with the LILO boot loader for cases when a new kernel does not boot at all.

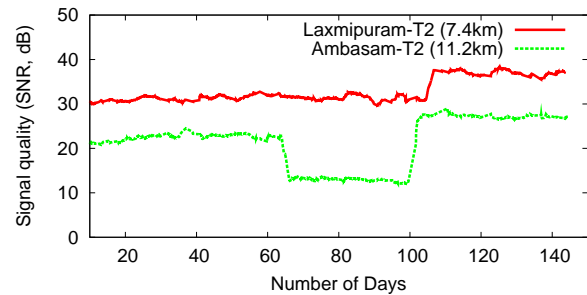


Figure 3: Signal strength variation on 2 links. There was an unexplained drop of 10dB on the Ambasam-T2 link for 25 days. Possible cause might have been temporary antenna misalignment.

5. PERFORMING FAULT DIAGNOSIS

In this section we first briefly describe the current status of our monitoring and diagnosis infrastructure. We then describe how real faults can be diagnosed using our framework.

5.1 Current Infrastructure

We are in the process of building the mechanisms described in section 4. Currently, we have implemented the monitoring infrastructure and have built a prototype power controller. We also use the hardware watchdog to recover from bad states. At the moment we do not have an SMS-based independent back channel and are currently evaluating different options for its implementation. We discuss our existing pieces and overall costs below.

Monitoring: Our monitoring infrastructure consists of log collection using PhoneHome and DTN running on every wireless node. The configuration of the nodes is performed by local admins using a web-based interface.

PhoneHome is designed to initiate connections from within the network (which is behind a NAT). It runs on every wireless router periodically (every 3 hours) and posts locally collected status information to a US-based server using HTTP. The posted information is collected and analyzed by scripts for identifying trends and producing graphs. An example use for monitoring variation in signal strengths is shown in Figure 3. *PhoneHome* also opens reverse SSH tunnels to enable remote experts to log into the Aravind nodes and perform diagnosis.

Power Controller: We have built a power controller for battery and power management. This micro-controller based programmable board supports solar panels and battery power (the next version will also support grid power), performs temperature sensing and peak-power tracking for charging the batteries. Importantly, it has the capability to report status information through the ethernet and serial ports. We use this controller for reporting panel voltages and current battery charge which can be used predict remaining uptime and battery lifetime. The controller can also be programmed to query the router board and reboot it if necessary.

Hardware watchdog: We also use the hardware watchdog on the processors to reboot the node from unrecoverable OS states. This is especially useful during system upgrades where we initialize the watchdog in LILO before booting with a new kernel. Subsequently if the new kernel fails to

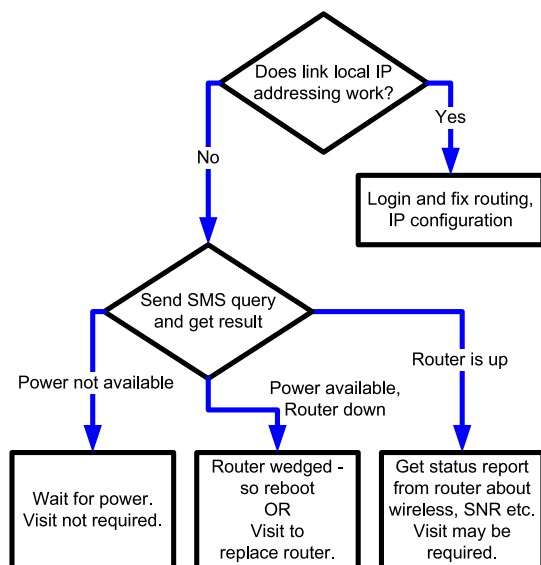


Figure 4: Partial flowchart for troubleshooting a link not working event.

boot properly and reset the watchdog, the board reboots automatically and LILLO loads a safe kernel configuration.

Costs: Excluding the cost of a tower or pole, each wireless node costs about \$450. This includes the router, a weather-proof enclosure, a 512 MB CF card, two wireless cards, up to two 24 dBi directional antennas, antenna cables, pigtailed and a PoE injector. If the node has no access to grid power, this cost increases by an additional \$275 for a 50W solar panel and \$45 for an 18 AH battery. To this base cost, we are adding \$70 for our power controller and an estimated \$50 for a cellphone and interfacing hardware. In India, SMS messaging is quite cheap. For example, \$3 monthly plans can be purchased where an SMS message containing 160 bytes costs about 2.5 cents. Assuming diagnostic parameters or test results can be represented as integers, 40 such values can be packed into one SMS, which we expect will be sufficient to provide a rich enough view of the node.

5.2 Typical Diagnosis Scenario

When a link from node A to node B stops working i.e. B is unreachable from A, there could be several reasons as seen in Table 1. The problem could be a simple routing misconfiguration, a temporary loss of power supply at the remote end B, an antenna misalignment at either end of the link or a hardware malfunction at the remote end B. A grid power supply shutdown does not require a physical visit to the wireless node, but if we can accurately identify an antenna misalignment, we would know that a physical visit is needed.

A basic flowchart for troubleshooting is shown in Figure 4. The first step is to try log in to B using the virtual local link addresses. If that is successful, then the problem is with routing or IP configuration. The next option is to use the SMS backchannel to query the remote router. If the SMS status reports that power is down at the remote end, then we can just wait for power to come back. However if the SMS reports that power is available but the router board is down, then we either need to reboot the board or undertake

a physical visit to replace broken hardware. If however, the board is up, the administrator can use the status report to isolate the problem further to non-functioning wireless radios, bad antenna cables, or antenna misalignments.

6. CONCLUSION

Simplification of diagnosis for rural WiFi networks is an important step in empowering rural administrators, building local capacity, and improving the operational sustainability of the system. In this paper we have presented a framework for designing networks with support for more accurate root cause fault diagnosis. The key ideas are building redundancy into hardware, software, and links for diagnosis such that some subsystems remain available even in the event of primary link failure and thus can still be queried. We show our progress to this end by describing our initial architecture and describing how faults can be diagnosed in this framework.

7. REFERENCES

- [1] Ashwini: Association for Health Welfare in the Nilgiris. <http://www.ashwini.org>.
- [2] Atheros. MadWiFi driver for Atheros Chipsets. <http://sourceforge.net/projects/madwifi/>.
- [3] K. Chebrolu, B. Raman, and S. Sen. Long-Distance 802.11b Links: Performance Measurements and Experience. In *ACM MOBICOM*, 2006.
- [4] CRCNet: Connecting Rural Communities Using WiFi. <http://www.crc.net.nz>.
- [5] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra. Implementing Delay Tolerant Networking. *Intel Research Berkeley Technical Report IRB-TR-04-020*, 2004.
- [6] Digital Gangetic Plains. <http://www.iitk.ac.in/mladgp/>.
- [7] R. Patra, S. Nedeveschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer. WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks. *NSDI*, 2007.
- [8] B. Raman. Channel Allocation in 802.11-based Mesh Networks. In *IEEE INFOCOM*, Apr. 2006.
- [9] B. Raman and K. Chebrolu. Revisiting MAC Design for an 802.11-based Mesh Network. In *HotNets-III*, 2004.
- [10] B. Raman and K. Chebrolu. Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks. In *ACM MOBICOM*, Aug. 2005.
- [11] RFC 3927: Dynamic Configuration of IPv4 Link-Local Addresses. <http://www.ietf.org/rfc/rfc3927.txt>.
- [12] Sayandeep Sen and Bhaskaran Raman. Long Distance Wireless Mesh Network Planning: Problem Formulation and Solution. *WWW*, 2007.
- [13] A. Sheth, S. Nedeveschi, R. Patra, S. Surana, L. Subramanian, and E. Brewer. Packet Loss Characterization in WiFi-based Long Distance Networks. *IEEE INFOCOM*, 2007.
- [14] L. Subramanian, S. Surana, R. Patra, M. Ho, A. Sheth, and E. Brewer. Rethinking Wireless for the Developing World. *Hotnets-V*, 2006.
- [15] The Aravind Eye Care System. <http://www.aravind.org>.