Wireless Mesh Network Management - Fault and Performance Diagnosis: A Survey

Master of Technology - Seminar Report

Vijay P Gabale
Computer Science And Engineering Department,
Indian Institute of Technology, Bombay,India
Under the guidance of Prof. Bhaskaran Raman

May 19, 2008

Abstract

A wireless mesh network grows organically as nodes are connected to each other, but they often lack centralized network management. Therefore, self management and self healing capabilities are the keys to the long term survival of these networks. Due to the inherent lossy nature of wireless networks, end users often experience unpredictable and unwanted performance. The causes of poor performance become non trivial to attribute in case of long-distance mesh network, especially in rural areas, where lack of expertise, erratic power conditions and intrinsic anomalies in wireless medium impede the speed of recovery.

These poor performance problems could be the result of link congestion, interference, MAC misbehavior, power outages, weak RF signals and the other reasons pertaining to the characteristic of actual deployment, like ACK level timeouts in case of long-distance mesh network. These causes often manifest themselves in terms of MAC level symptoms like retransmissions or user level experiences like intermittent connectivity. But the remedial actions make it necessary to identify and categorize the root-level faults. In dense enterprise wireless architectures, the deployment of multiple wireless monitors, in the form distributed data collection network which pushes data up to a central server for diagnosis, is necessary to characterize the network whereas in long-distance mesh networks, independent per node control mechanisms and remote monitoring can be employed to tackle performance problems. This survey looks at the various angles of network management in terms of performance and fault diagnosis in wireless mesh networks.

1 Introduction

802.11 wireless networks have become ubiquitous owing to cheap wireless interfaces, unlicensed spectrum and inherent convenience of untethered computing. 802.11 devices, in turn, can be connected to each other to form mesh network which are easy to deploy and maintain, scalable and reliable. A wireless mesh network grows organically as nodes are connected to each other, but they often lack centralized network management. In typical enterprise office setting or university buildings, wireless mesh networks result in dense deployment of 802.11 devices whereas in case of providing connectivity to remote areas like villages, wireless nodes are spaced quite distant from each other and use high gain directional antennas to establish communication. Due to distributed nature and lack of centralized administration, the self management and self healing capabilities are the keys to the long term survival of these networks.

With the growing popularity of wireless network in both enterprise as well as long-distance scenarios, users are expecting for increased performance, scalability and reliability. However 802.11 networks have to face harsh realities like unpredictable ambiance, packet losses, area coverage susceptibilities and other such performance issues. These issues are either direct or by-products of dense deployments in enterprise networking, interference with 802.11 and non 802.11 devices operating in the same frequency range or RF effects of the medium. As a result, these anomalies, as they are called because of their unexpected and abnormal behaviour, manifests themselves quite frequently, in poor performance and loss of productivity. Poor performance is often rendered as decrease in throughput or longer response time. The rectification of these anomalies requires extra support to identify and eliminate them by taking reactive measures. Thus, with low cost connectivity feature, 802.11 brings the inherent anomalous behavior of wireless medium to wireless mesh network and the very capabilities of being self-manageable become vulnerable to frequent faults occurring in the network. Network management becomes especially challenging in long distance (rural) wireless mesh environments since there could be manifold reasons for a specific fault and the attribution of a fault to a spe-

cific root cause is difficult.

Thus, a vulnerable network makes it necessary to diagnose the faults and take remedial actions as early as possible. This results into robust, resilient and well-planned network that gives enhanced performance to users with better connectivity. The issues involved, however, are identifying the exact causes of faults across layers, exploiting physical layer information, taking appropriate remedial actions and automizing these tasks.

Most of these issues and problems that we encounter in mesh networks are due to the fact that systems are not designed or deployed with support for easy diagnosis built right from start. When a wireless link or wireless node goes down, the node might become unreachable or at worst the whole network may come to grinding halt. Until we figure out what has gone wrong, system downtime increases and users loose their productivity. Without tools or appropriate measures to characterize these failures, it requires several hours of time and energy in repair work.

In case of dense enterprise networks, the challenge lies in characterizing the wireless behavior through monitoring the network, quantifying physical layer parameters and categorizing causes to specific faults. The monitoring infrastructure is intended to provide answers to questions like how many concurrent transmissions were there?, how is per packet signal strength varying over time?, are there any non reachable nodes? etc. The answers to these questions help identify hidden terminals, antenna misalignment and RF holes (and similar problems) respectively.

While we can deploy multiple sensors to perform fault diagnosis in dense enterprise networks, the problem over long-distance links through remote (rural) areas becomes critical. Typically, nodes are spread distant from each other, expertise to solve simple networking problems may not be available and it incurs personal visit to figure out the fault at distant nodes. The cause of faults occurring could be as diverse as interference due to water pump in a farm-field or damage of router boards due to power spikes. The failures due to poor power quality could be rife in rural areas. The task lies in arriving at a remote monitoring solution, taking decision of whether to have pull based architecture; where a daemon running on one machine queries others or push based architecture; where every node pushes pertaining data to central server and incorporating additional software and hardware components to make recovery automatic whenever possible.

This survey attempts to provide answers to questions like, What techniques should be in progress to get insights of the possible causes? how do we categorize the causes of a fault? If the diagnosis says that interference is the problem, then how do we go about fixing it? What is the mechanism to apply or action to take? The organization of the report is as follows: Section 2 classifies and explores various techniques proposed to handle fault diagnosis. Section 3 elaborates on how to handle specific faults in wireless mesh networks. Section 4 briefly states the system components required for various techniques. Finally, the survey concludes with the possible scope for future work and conclusion in section 5 and section 6 respectively.

2 Survey of existing techniques

To provide answers to these questions in systematic manner, network management and fault diagnosis research has been undertaken in the domain of small and large scale Enterprise WiFi deployments. Comparatively very fewer efforts have been made so far towards long-distance mesh network management and root cause analysis though. This section categorizes the approaches taken so far for enterprise and long-distance fault diagnosis into five major appoaches: (a) offline diagnosis on network traces (b) online anomaly detection system (c) simulating expected behavior to compare with observed behavior (d) a daemon working as a part of the node (e) system incorporating additional (redundant) software and hardware components in the node.

2.1 Offline diagnosis

In this approach, multiple monitors are deployed close to clients and APs. Each monitor individually collects data, bundles them over a time period and sends them to a central machine. The central machine, equipped with inference measures, collects these traces. It applies certain techniques to synchronize and unify these traces. To synchronize traces received across clients, either beacon frames, which carry unique 64-bit timestamp, are used or certain unique reference frames are identified which help order the frames in the sequence in which they were transmitted.

Although the dense deployment of monitors is expected to capture all the ongoing transmissions, some transmissions sometimes elude the monitors. To infer about these losses and to have comprehensive trace of wireless activities, Finite State Machines (FSM) are developed for wireless protocols and are applied over the traces. Custom inference techniques are then used to infer frames missed by monitors themselves or to get rid of duplicates. Eventually a single unified trace characterising entire wireless behavior over a period of time is built. Figure 1 shows the the technique pictoreally. [1] uses this approach where around 150 passive radios collect traces and a complete wireless behavior is reconstructed in terms of records and conversations. Inference techniques are then applied to detect concurrent transmissions for calculating number of frames collided. The motive behind this system is to produce a precisely synchronized global picture of physical, link, network and transport layer activities for analysis of large 802.11 networks. The system gives deeper insights about the fraction of beacon and ARP traffic comprising of overall traffic in their network, probability of interference given simulate-

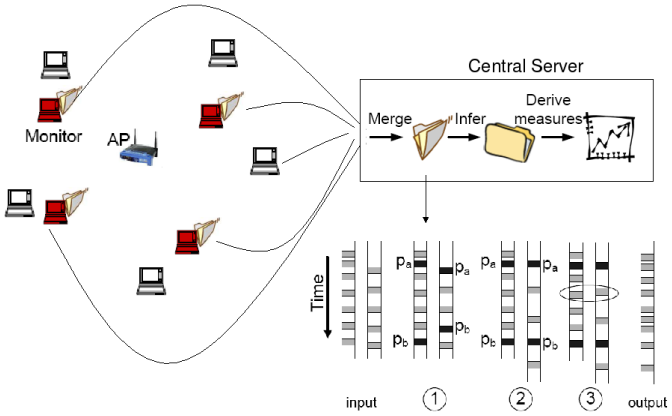

Figure 1: Offline diagnosis framework

neous transmissions and overprotective 802.11g devices.

2.2 Online diagnosis

While the offline approach works well to characterise the entire network, we require online and dynamic network setting to detect faults as soon as they take place. The online diagnosis approach also involves deploying multiple monitors close to clients and APs to capture transmission frames. Nodes periodically sample parameters like noise floor, signal strength etc and forward them to central inference engine which makes decisions dynamically. The inference engine running at this central machine outputs probable faults like RF holes depending on known spatial locations and link asymmetries etc.

[2] employs this approach where faults like hidden terminals, capture effect, noise are artificially replicated in the network and the behavior of network under a known fault is characterized in terms of thresholds. Whenever certain parameters cross pre-defined threshold, the corresponding fault is triggered as the possible reason of anomaly. This study very elegantly characterizes faults like hidden terminals, capture effects and non 802.11 interference with the help a small test bed and using custom techniques to introduce noise.

2.3 Simulation

In previous two subsections, we saw how online and offline diagnosis can be done. But, simulation could also be a viable option where we do not require to set up testbed and take measurements. A simulator could be used to figure out the deviation of observed behavior from expected behavior. To drive the simulation study, as a preliminary step, data or traces are collected by observing transmissions in a mesh network, inconsistencies are removed from traces, and then these traces are fed to simulator to extract expected behavior. Any significant deviation of observed behavior from expected behavior is categorized as anomaly. A suitable efficient search algorithm with the help of decision tree is used to come up with faults that best matches the observed behavior.

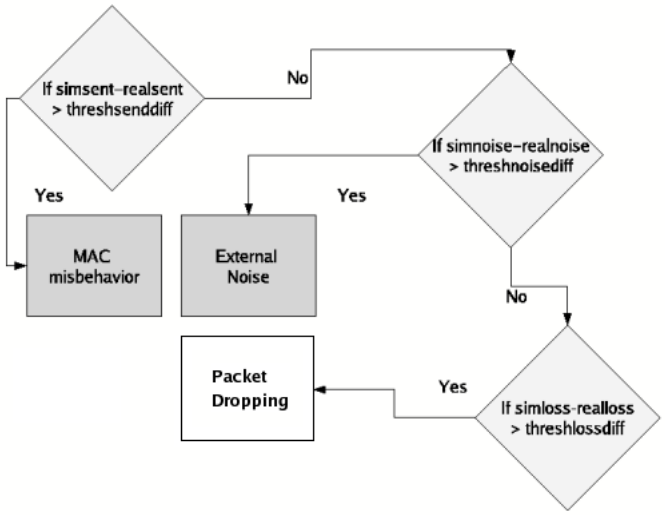

Figure 2: Decision tree for detecting faults in simulation

Troubleshooting[3] uses this technique to categorize faults as packet dropping at the receiver, excessive transmissions resulting in link congestion, external interference or MAC misbehavior.

The search space for faults for best match is high-dimensional due to combinations of faults. But, to make search efficient, we can take advantage of the fact that different types of faults often change one or few metrics. For example, external noise sources increase noise experienced by its neighboring nodes but do not increase the sending rate, and therefore can be differentiated from MAC misbehavior and packet dropping at hosts. In [3], a decision tree is build based on this predicate as shown in figure 2. The figure shows how the difference between simulated parameters and observed parameters can be used along with parameter specific threshold to classify the faults. Though simulations are cost-effective, they may not capture the intrinsic wireless behavior and hence results obtained often deviate from real-world measurements.

2.4 Daemon running as the part of the node

Unlike in offline and online approach, a part of duties of central unification client can be delegated to individual nodes. In this approach, an application, as a part of diagnostic system, runs on each node. The application is configured to take certain actions in response to events occuring in the network. For example, in terms of disconnection, the application (called client conduit in architecture and techniques[4] paper) turns the node into an AP to find out nearest client connected to real AP to transmit its log to detection system. The application, then, may prompt two nodes to form ad hoc network using techniques like multinet which multiplexes a single wireless card to be part of two networks at a time.

The client conduit[4] technique can be further used to locate disconnected clients and rogue APs. This mechanism is

explained in section 3.

2.5 Software and hardware redundancy

This approach is particularly suitable for long-distance links. Through Arawind and AirJaldi[5] networks, it is experienced that software and hardware failures are quite rampant in long-distance (rural) networks. Thus we require independent control mechanisms to curb software and hardware component failures inside a node. As a part of this structure, a daemon runs on each node to supply the parameters regarding the health of node and links to neighboring nodes to a remote server. The remote server, in turn, runs certain inference techniques to diagnose faults of poor performance like packet loss due to interference or external noise etc. Software and hardware watchdogs as the part of the node are built to mitigate problems due to power quality and system malfunction events. Backchannels can be used in case of primary link failure to know the health of node. One way to realize the backchannels is to use Short Messaging Service (SMS) by keeping Mobile phone inside the node.

Beyond pilots[5] uses these techniques to resolve software and hardware failures that they experienced in their deployments in India. These techniques are further elaborated in section 3.

3 Fault Diagnosis in wireless mesh networks

Previous section categorized the techniques to deal with fault diagnosis. This section exemplifies how different faults in the network can be detected using these techniques. Here, the fault diagnosis examples are explained in terms of symptoms observed, possible causes of the fault, techniques to be used and actions to be taken to mitigate these faults. This survey identifies two broad network types, dense enterprise network and long-distance mesh networks for fault diagnosis. The techniques for the two differ substantially, so as the faults occuring. Separate subsections are dedicated to these networks which are followed by a short description of how to get the parameters required to quantify the performance measures.

3.1 Enterprise networks

Starting with enterprise wireless network, it consists of wireless deployments in office, university building, home or in commercial campus. The network comprises of densly deployed stationary or mobile clients in close vicinity of access points spread over buildings. Users here experience numerous problems such as intermittent connectivity, poor performance, lack of coverage, authentication failures etc. This subsection provides insights into how some of these problems can be resolved.

3.1.1 Connectivity problems

- Symptoms: Intermittent connectivity and total failure
- Causes: Weak RF signal, Lack of signal, unpredictable ambiance, obstructions
- Techniques: The key lies in tracking the received signal strength values as they depict the coverage in a specific region. But the problem here is that, once a client gets detached from the network due to loss of connectivity, how can it possibly convey the problem to central administrator. The client conduit protocol[4] mentioned in Architecture and Techniques paper tackles this ambiguity as follows:

  1. The Diagnostic Client on the disconnected client (disconnected from AP) configures the machine to operate in promiscuous mode. It scans all channels to determine if any nearby client is connected to the infrastructure network.
  2. This newly formed AP at disconnected node broadcasts its beacon like a regular AP.
  3. Every client in the network has to perform active scanning and while doing this, when the client receives this beacon, it sends probe message.
  4. Disconnected station becomes normal station again and sends reply message.
  5. Connected node starts ad hoc network with disconnected client via Multinet. (The connected client first performs authentication through certificates. Also the number of times this can be done is constrained to refrain connected client from wasting its resources for helping disconnected client)

  The traces of disconnected clients are then conveyed to central server. Double Indirection for Approximating Location (DIAL) protocol is then used to locate the disconnected client. This is done by (a) measuring signal strength values for connected client which acts as intermediary and then (b) using parameters of disconnected clients.

- Actions: Relocate AP, adjust transmission power for better coverage

3.1.2 Detecting rogue APs

- Symptoms: Authentication failure, unauthorized access
- Causes: Disgruntled employees, unawareness on the part of the user
- Techniques: Rogue APs are unauthorized APs that get connected to an Ethernet tap in an enterprise or university

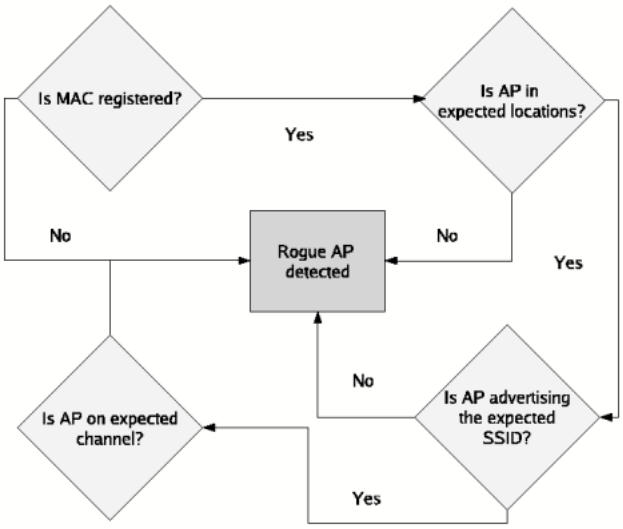

Figure 3: Flow diagram to detect rogue AP

network; such APs can result in security holes and unwanted RF and network load. It is required to locate the rogue APs as soon as possible before they cause destruction. The architecture and techniques[4] employs following apporach:

1. For each AP that a node detects, it sends a four tuple: MAC address, SSID, channel, RSSI to the Diagnostic (central) Server, this four tuple uniquely identifies an AP in a particular location and channel. The AP MAC address can be determined using beacon frames. These are then mapped with location databases to calculate the current position of (detect) rogue AP. Figure 3 shows decision tree to detect rogue access points. Such a decision tree makes use of the parameters collected from the nodes and location database available at the central node.
2. Active scanning on the part of the client can also result in detecting an AP working on unexpected channel or through channel overlaps.

- Actions: Authentication using certificates

3.1.3 Hidden terminal

- Symptoms: Degraded performance, lower throughput
- Causes: One transmitter not able to hear other transmissions to the same receiver (the hidden node problem[2]), heterogeneous transmit power
- Techniques: In Mojo[2], hidden terminal anomaly is deliberately arranged and then quantified in terms of goodput and percentage of frames collided. It is observed that about 40% of the frames collide when hidden node terminal anomaly is present. This number is termed as threshold for hidden node terminal. To detect an instance of this fault, multiple monitors capture on-going activities. For every pair of adjacent data frame transmissions, it is checked whether there are more than (or close to) 40% (overlapping) concurrent transmissions directed to same node. If they are present, then the degradation in performance is due to hidden terminal effect.
- Actions: Step up transmit power of hidden terminals or use RTS/CTS

3.1.4 Capture Effect

- Symptoms: Degraded performance, lower throughput
- Causes: High transmission power station acquiring and making unfair use of channel
- Capture Effect: In wired communication environment, the packets will be considered as collisions if two packets arrive in the same station at the same time. Even though the received power for one of the packets is much larger than the other one, the station still takes all the packets as collisions. However, as it turns out for the wireless communication network, even though more than two packets arrive in the same station at the same time, the packet with high Signal to Noise Ratio (SNR) (greater than predetermined threshold) still can be received successfully. Thus the corresponding station will always capture the channel. The natural question is Why would two stations that are in the range of each other transmit concurrently when both use the CSMA/CA protocol? The answer is that contention window is set to min after each successful ACK (Acknowledgment) and backoff interval is selected based on this number. Also it takes only 25 microseconds to clear channel assessment.
- Techniques: In [2], the catpure effect anomaly is deliberately arranged and then quantified in terms of goodput and percentage of frames collided. It is observed that about 5% of the frames collide when capture effect anomaly is present. This number is termed as threshold for capture effect. To detect an instance of this fault, Multiple monitors capture on-going activities. For every pair of adjacent data frame transmissions, it is checked whether there are more than (or close to) 5% (overlapping) concurrent transmissions directed to same node. If they are present, then the degradation in performance is due to the capture effect.
- Actions: This anomaly results due to the mismatch between transmit power and receiver sensitivity across stations and can be mitigated by adjusting transmit power to give fair access to medium.

3.1.5 Non 802.11 device interference

- Symptoms: Retransmissions at the MAC layer, no concurrent transmissions
- Causes: Since non 802.11 devices do not follow media access protocol and since they cause channel interference, they result in 802.11 frame corruption, excessive back offs and frequent retransmissions.
- Techniques: To detect erratic noise erruption, we need to dynamically track the signal energy level present at a station. As soon as any unwarranted spike is detected, it could be the reason of noise interference by non-802.11 devices. Thus, for a time period, termed as EPOCH INTERVAL in [2], set by administrator, noise floor value is sampled and a moving average is maintained. As soon as the window crosses threshold (set based on observations), a fault is triggered as non 802.11 device interference.
- Actions: Typically the non 802.11 device is identified and removed, the other measure could be changing the operation frequency for a node(channel).

3.2 Long distance networks

As compared to enterprise fault detection seen in previous subsection, in long-distance mesh network, the remote diagnosis and repair of faults like interference, intermittent connectivity become challenging to address because

- Physical visits are costly.
- Remote locations could sometimes become inaccessible.
- Trained personnel may not be available.
- Poor power quality could be the real culprit.
- Deploying network-wide-passive-monitoring infrastructure becomes infeasible.

For these kinds of network, support is required in collecting network topology and wireless configuration data at a central location, measuring signal strength and noise variations and measuring the wireless error rate and throughput in the network. Also without visiting the locations personally for annoying reasons like "it was only needed to reboot", it should be possible either to take remedial action like "automatic rebooting", "automatic shutdown on power spike" or make inferences like "Board needs to be replaced" so that visits can be scheduled and result to be fruitful.

The problems present in these networks often render themselves as remote node unreachable or primary link failure. Let us look at the various faults that occur in these networks and how we can resolve them.

3.2.1 Connectivity problems

- Symptoms: Remote node NOT Reachable
- Causes: IP address misconfiguration, routing misconfiguration, primary link fails, power shutdown at remote node, a board failure, malfunctioning wireless card

  Here we need to detect the exact cause among these possible causes. To achieve this, we should be able to query the node through some other means like backchannels (explained later). If this succeeds, it makes sure that the node is up and working. Then the reasons for problem could be configuration issues which can be solved once logged into the node. If the attempt to connect to the node fails, then we should have some mechanism which queries the 'on board but independent' equipments working inside the node. This mechanism in turn, gives us back a status report for the node. This should assist us in predicting whether there is power shutdown or board failure or a software malfunction. Figure 4 shows the flow chart of this decision process and how we can arrive to a probable conclusion.
- Techniques: Independent back channels

  1. Link local IP addressing[6]: This technique is typically used to establish a connection with remote host whose interface is misconfigured. A link local IP addressing enabled host can automatically configure an interface with an IPv4 address within the 169.254/16 prefix that is valid for communication with other devices connected to the same physical (or logical) link. Using this feature, we can log into the other host and can resolve misconfiguration problems by setting certain parameters like IP address and gateway address.
  2. SMS Request Reply: In case of node failure, link-local IP addressing fails. As an alternative, a mobile phone can be kept inside the router and is connected to the board such that, it periodically gets information regarding whether the components are working properly or not. A remote SMS query is sent in case of primary link failure (we do not know reason yet why the node is failed) to know the exact problem, that is whether power supply is failed or board is failed. The response message contains parameter regarding the board state, power supply state etc. This helps predict and schedule the remote trip.
- Actions: Reconfigure interfaces using link local IP address technique, replace boards, power supply etc.

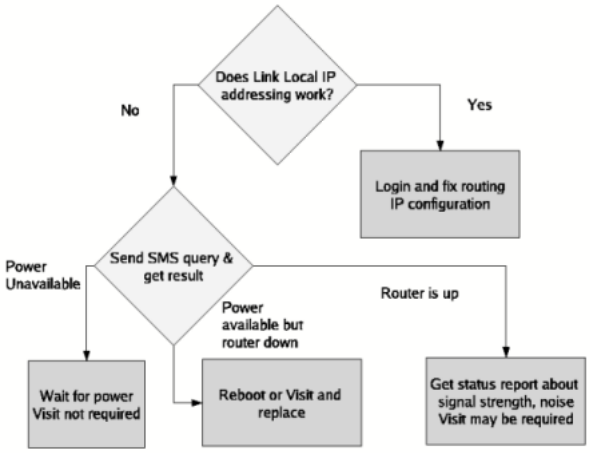

Figure 4: Troubleshooting a link

3.2.2 Software and hardware failures

- Symptoms: Node suddenly goes down, node does not respond on trying to connect over the primary link
- Causes: Damage of power supplies or router boards, low voltages leave router in wedged state, fluctuating voltage causes frequent reboots; which damage the on board CF memory card through writes during reboot, prevents optimal charging of batteries
- Techniques: Software and Hardware watchdogs, Power controllers, Read only boot loader

  Following solutions are designed to increase component reliability in the face of bad power in Beyond Pilots[5] paper:

  1. Low Voltage Disconnect: prevents over discharge (over discharge reduces lifetime) by disconnecting load when battery voltage drops below a threshold and prevents router from being powered by a low voltage source
  2. Power Controller: provides quality and regulated voltage to nodes, manages charging and discharging, can be used in aid with solar power
  3. Read-Only Boot-Loader: does not write on CF cards while rebooting and prevents getting corrupted due to frequent and unexpected reboots occuring in the network

- Actions: Build software and hardware redundancy in the system

3.3 Measures

In last two subsections, we saw how we can apply different techniques to characterize various faults. But to quantify these faults, we need to measure underlying parameters like per packet signal strength values, noise floor or sequence numbers in the packet. To extract these values, low level access to physical layer header information is required in terms 802.11 frames where we can capture the per packet signal strength, noise values and can also determine whether checksum was passed or not. Following examples give glimpse of how these parameters can be obtained:

- Open source MADWIFI driver periodically calibrates the noise floor (benchmark for CCA) into a hardware register. This hardware register can be sampled to get noise floor value experienced by the client.
- We can either extract the signal strength field from the prism header for every beacon frame or driver can be changed to directly pass the per packet values from kernel level to user level.
- Packet loss can be computed from sequence numbers of packets received. Several correlation techniques can be applied to figure out the whether there exist any interesting relation between the packet losses.
- Each frame contains a unique 64-bit timestamp (related to sender) and the fact that propagation time is negligible can be exploited to find out the number of concurrent transmissions.

3.4 System Components

In earlier sections, we saw various techniques in specific to apply to detect faults. These techniques and the corresponding metrics can be realized in high level abstraction as one or more system components. These components that employ the certain techniques are a daemon that runs on every node to collect node specific information, inference engine at a server which collects traces from clients and optional back channels to query nodes in case of primary link failure.

- Monitoring system: The daemon keeps track of number of missing frames, signal strength variations, link qualities to neighboring nodes. Additional software watchdogs keep an eye on network parameters misconfiguration, viruses eating bandwidth and predicting of network hardware and software component failures. The software watchdogs occasionally recover the routing daemon (restart it). The hardware watchdogs track the board status, power level and battery health.

  [5] mentions numerous experiences of using PhoneHome as their monitoring system. The system helped them maintain reachability information which can alert local staff of network failure. The kernel logs, obtained through the system, helped them further to diagnose problems like - under low power only one card could remain active even though the router had two cards present.

This continuous monitoring system also allowed them identify faults like antenna misalignment.

- Inference engine: When the data collected through monitoring system arrives at a central node, which is specially configured to be able to apply costly search algorithms, the node goes over the data collected and makes predictions or draws conclusions from variation in various parameters. Finite State Machines, as mentioned in section 2, are incorporated into the inference engine to cope with monitoring system inabilities like missed packets in a conversations.
- Optional back channel: These come handy when primary link goes down due to reasons explained in previous subsections. Optionally we can have shell access or can open reverse SSH tunnels to execute commands at the remote node to find the exact problem. SMS request reply mechanism, which has become affordable in India, can be coupled with the nodes (as done in [5]) and can be applied to detect the node health.

4 Future work

Though the faults faced in enterprise and long-distance deployments have been quantified and characterized, there is a need to develop a comprehensive network monitoring and inference tool for both enterprise and long-distance networks. Also, though sophisticated tools do exist to detect individual faults, a single tool could serve to be of much use to characterize the performance of network at scale.

The Beyond Pilots[5] paper has manifold ingeneously developed solutions to tackle problems in long-distance mesh networks, but the paper does not quantify the efficacy of the techniques in terms of performance improvement. Thus experiments must be designed and tested for the techniques like software and hardware redundancies to quantify the performance of enhanced network.

In case of rural areas, where we need to employ local expertise, a user friendly GUI could serve to be of great help for managing and maintaining the network locally.

The procedure of remedial actions can be made automatic. To give an example, after detecting capture effect[2], we should automatically be able to set the transmit power of both stations appropriately to give them fair access.

5 Conclusion

The intrisic wireless medium characteristics and by-products of dense and long-distance wifi deployments give rise to manifold performance problems. This survey classifies the techniques to solve performance problems in five different categories: (a) offline diagnosis (b) online diagnosis (c) diagnosis through simulation (d) system with per node daemon and (e) redundancy in terms of hardware and software component across two different networks: enterprise and long-distance. It also delves into the possible faults and their remedies that are encountered in enterprise and long-distance mesh networks. Some of the techniques mentioned, are developed specially to characterize the entire wireless behavior in a campus building whereas others deal with quantifying thresholds for certain faults.

The problem becomes non-trivial in case of long-distance networks where hardware failures become the cause of concern. Various faults like RF holes, hidden terminals, capture effects, interference, power failures etc are studied in terms of symptoms experienced by the users, the possible causes, how do we categorize the causes and select the one that best matches the anomaly and what are the possible actions we can take to mitigate the faults. The metrics to quantify the causes are also listed along with the major components of the framework required to gather the information. The survey is completed with the need to make comprehensive, automated, user friendly tool that can monitor remote network barring failures and can help manage the network locally.

References

[1] Yu-Chung Cheng, John ellardo, and Peter Benko. Jigsaw: Solving the Puzzle of Enterprise 802.11 Networks. In *SIGCOMM*, 2006.

[2] Anmol Sheth, Christian Doerr, Dirk Grunwald, Richard Han, and Dougla Sicker. Mojo: A Distributed Physical Layer Anamoly Detection System for 802.11 WLANs. In *MOBISYS*, 2006.

[3] Lili Qiu, Paramvir Bahl, Ananth Rao, and Lidong Zhou. Troubleshooting Wireless Mesh Networks. In *SIGCOMM*, 2006.

[4] Atul Adya, Paramvir Bahl, Ranveer Chandra, and Lilli Qiu. Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks. In *MOBICOM*, 2004.

[5] Sonesh Surana, Rabin Patra, and Sergiu Nedevschi. Beyond Pilots: Keeping Rural Wireless Networks Alive. In *USENIX NSDI*, 2008. To appear.

[6] Dynamic configuration of IPv4 Link-Local Addresses. http://www.ietf.org/rfc/rfc3927.txt.

[7] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Analyzing the MAC Level Behavior of Wireless Networks in the Wild. In *SIGCOMM*, 2006.

[8] Sonesh Surana, Rabin Patra, and Eric Brewer. Simplifying Fault Diagnosis in Locally Managed Rural wifi networks. In *SIGCOMM NSDR*, 2007.

[9] Kameswari Chebrolu, Bhaskaran Raman, and Sayandeep Sen. Long-distance 802.11b Links: Performance Measurements and Experience. In *MOBICOM*, 2006.

Appendix

Comparison table of techniques

Work	Methodology/Category	Faults/Problems Handled	Metrics/Techniques Used	Feature
Mojo	Online physical layer anomaly detection system	Hidden Terminal Capture Effect Noise Signal strength variations	Noise floor RSSI Number of concurrent transmissions	Inference using Finite state machine of MAC protocol
Jigsaw	Offline reconstruction of global cross-layer viewpoint	Degree of contention Effect of 802.11b clients in 802.11g network	Number of concurrent transmissions P(Interference | simultaneous transmissions)	Massive monitoring deployment comprising of 150 passive radios to characterize entire wireless behavior
MAC Level Behavior in Wild	Offline unification of MAC level frames	Number of stations contending	Number of concurrent transmissions	Formal Language method to construct Finite State Machine
Architecture & Techniques for Diagnosing Faults	Per Client Daemon & a Server to collect data from Clients	Locate RF holes Locate Disconnected Clients Wireless Performance Detecting rogue APs	Client Conduit Protocol DIAL Loss rate Packet Delay using EDEN	A novel architecture comprising of Diagnostic Client that runs on each node & Diagnostic server
Troubleshooting Wireless Mesh Networks	Simulation of Wireless Mesh Network driven by deployed network traces	Packet dropping Link congestion External noise MAC misbehavior	Packet loss Inconsistency graph Measuring the size of Contention Window	Efficient search algorithm to find best matching fault
Beyond Pilots	Software & Hardware redundancy with provisions for remote monitoring of long-distance mesh network	Failure due to bad power quality External Interference Antenna misalignment or damaged pigtail connectors Network Partition	Packet loss RSSI variations Push-based “Phone-Home” application Low Voltage Disconnect Link Local IP addressing Cell phone backchannels	Software & Hardware watchdogs (Independent Control Mechanisms) to take care of erratic behavior with Backchannels