Fault detection and diagnosis over a Wireless Network (Using trace-driven simulation)

Ibitoye Akinfolajimi Akinrinlola

Department of Computer Science Lagos State Polytechnic Ikorodu, Lagos, Nigeria.
Email: mideanddee@hotmail.com.

Accepted 7 September, 2015

Given a wireless network and a graph modeling the node connection topology, we address the problem of fault diagnosis of nodes. When the external point of access to network information is at a lower layer of the ISO/OSI protocol stack, the problem is trivial. However, as often occurs, a user is able to access the network at the application layer: this implies that the available observation of the network status is considerably restricted, and solving the diagnosis problem is not trivial. This paper states the necessary and sufficient conditions for being able to detect which node is faulty, and propose a diagnosis algorithm, a system that employs online trace-driven simulation on the basis of diagnosability definitions and theoretical studies developed for timed and hybrid automata in the computer science community. We assumed that our interface with the network is a gateway node that accepts ping commands (control input), and replies with pong responses after a time delay that we can measure (observed output). Network management in wireless networks is key to efficient and reliable network operation. This paper focuses on a part of the general network management problem, namely fault detection, isolation, and diagnosis. It proposes a system that employs online trace-driven simulation as a diagnostic tool for detecting faults and performing root cause analysis.

Key words: Wireless, network, topology, gateway.

INTRODUCTION

A process that is used to reconfigure or help in detecting the different types of problems and give solutions to correct them is called troubleshooting. Troubleshooting or fault detection in a wireless network is completely based on the problems that are technical in nature and reduce the distribution or the spreading of the technical faults in the whole operating system is called window troubleshooting. As we know that wireless, technology becomes very famous among all the aspects of different technologies of computer networking of present era. Wireless networks really provide the convenient and easy approach to communications between different areas. So, there are several technical faults in the wireless networking that can slow down the working of the wireless media. So, for detecting and resolving these technical problems in the wireless network is called as wireless network troubleshooting. Basically wireless network troubleshooting is used to detect the source of the problem and resolve the problem from roots and maintain the working of the whole wireless networking media.

LITERATURE REVIEW

History of Wireless Network and Fault Detection

Wireless network refers to any type of computer network that is not connected by cables of any kind. It is a method by which telecommunications networks and enterprise (business), installations avoid the costly process of introducing cables into to a building, or as a connection between various equipment locations. Wireless telecommunications networks are generally implemented and administered using a transmission system called radio waves. This implementation takes place at the physical level, (layer), of the network structure.

Devices which use radio frequency transmissions to replace medium-distance wired connections (such as Ethernet cables), to create what is sometimes called a WAN (wireless area network), or employed simply to replace short-distance cabling between digital devices. The most common current wireless network protocols are
in the 802.11x family, sometimes referred to as Wi-Fi (Wireless Fidelity), and have a range of hundreds of feet. The shorter-range Bluetooth specification is intended for communications over tens of feet.

Wired networks are protected by physical security mechanisms associated with the buildings in which they are housed. Signals broadcast by wireless devices can travel far beyond the confines of any structure, and so must be protected in other ways. Both Wi-Fi and Bluetooth specifications include security elements, and most if not all of these elements are implemented by manufacturers of devices with wireless capabilities. However, these security measures are often turned off by default. Wireless devices without such security measures enabled can be very insecure indeed. Since wireless devices also tend to be portable ones, they also have the added security risk of being more easily lost or stolen.

Types of wireless connections

Wireless PAN

Wireless Personal Area Networks (WPANs) interconnect devices within a relatively small area, generally within a persons reach. For example, both Bluetooth radio and invisible Infrared light provides a WPAN for interconnecting a headset to a laptop. ZigBee also supports WPAN applications. Wi-Fi PANs are becoming commonplace (2010) as equipment designers start to integrate Wi-Fi into a variety of consumer electronic devices. Intel "My WiFi" and Windows 7 "virtual Wi-Fi" capabilities have made Wi-Fi PANs simpler and easier to set up and configure.

Wireless LAN

A wireless local area network (WLAN) links two or more devices using a wireless distribution method, providing a connection through an access point to the wider internet. The use of spread-spectrum or OFDM technologies also gives users the mobility to move around within a local coverage area, and still remain connected to the network.

a. Wi-Fi: "Wi-Fi" is a term used to describe 802.11 WLANs, although it is technically a declared standard of interoperability between 802.11 devices.
b. Fixed Wireless Data: This implements point to point links between computers or networks at two distant locations, often using dedicated microwave or modulated laser light beams over line of sight paths. It is often used in cities to connect networks in two or more buildings without installing a wired link.

Wireless MAN

Wireless Metropolitan Area Networks are a type of wireless network that connects several Wireless LANs. WiMAX is a type of Wireless MAN and is described by the IEEE 802.16 standard.[1]

Wireless WAN

Wireless wide area networks are wireless networks that typically cover large outdoor areas. These networks can be used to connect branch offices of business or as a public internet access system. They are usually deployed on the 2.4 GHz band. A typical system contains base station gateways, access points and wireless bridging relays. Other configurations are mesh systems where each access point acts as a relay also. When combined with renewable energy systems such as photo-voltaic solar panels or wind systems they can be stand alone systems.

MOBILE DEVICES NETWORKS

With the development of smart phones, cellular telephone networks routinely carry data in addition to telephone conversations:

1. Global System for Mobile Communications (GSM): The GSM network is divided into three major systems: the switching system, the base station system, and the operation and support system. The cell phone connects to the base system station which then connects to the operation and support station; it then connects to the switching station where the call is transferred to where it needs to go. GSM is the most common standard and is used for a majority of cell phones.[6]
2. Personal Communications Service (PCS): PCS is a radio band that can be used by mobile phones in North America and South Asia. Sprint happened to be the first service to set up a PCS.
3. D-AMPS: Digital Advanced Mobile Phone Service, an upgraded version of AMPS, is being phased out due to advancement in technology. The newer GSM networks are replacing the older system.

USES OF WIRELESS NETWORK

Wireless networks continue to develop, usage has grown in 2010. Cellular phones are part of everyday wireless networks, allowing easy personal communications. Intercontinental network systems use radio satellites to communicate across the world. Emergency services such as the police utilize wireless networks to communicate effectively. Individuals and businesses use wireless networks to send and share data rapidly, whether it be in a small office building or across the world.

Another use for wireless networks is a cost effective means to connect to the Internet, in regions where the
A telecommunications infrastructure is both poor and lacking in resources, typically in rural areas and developing countries. Compatibility issues also arise when dealing with wireless networks. Different devices may have compatibility issues, or might require modifications to solve these issues. Wireless networks are often typically slower than those found in modern versions of Ethernet cable connected installations.

A wireless network is more vulnerable, because anyone can intercept and sometimes divert a network broadcasting signal when point to point connections are used. Many wireless networks use WEP - Wired Equivalent Privacy - security systems. These have been found to be still vulnerable to intrusion. Though WEP does block some intruders, the security problems have caused some businesses to continue using wired networks until a more suitable security system can be introduced. The use of suitable firewalls overcome some security problems in wireless networks that are vulnerable to attempted unauthorised access.

Environmental concerns and health hazard

Starting around 2009, there have been increased concerns about the safety of wireless communications, despite little evidence of health risks so far.[7] The president of Lakehead University refused to agree to installation of a wireless network citing a California Public Utilities Commission study which said that the possible risk of tumors and other diseases due to exposure to electromagnetic fields (EMFs) needs to be further investigated.[18]

HISTORY OF WIRELESS COMMUNICATION

Figure 1. An embedded RouterBoard 112 with U.FL-RSMA pigtail and R52 mini PCI Wi-Fi card widely used by wireless Internet service providers (WISPs).

In 1970 Norman Abramson, a professor at the University of Hawaii, developed the world’s first wireless computer communication network, ALOHAnet, using low-cost ham-like radios. The system included seven computers deployed over four islands to communicate with the central computer on the Oahu Island without using phone lines.[3]

"In 1979, F.R. Gfeller and U. Bapst published a paper in the IEEE Proceedings reporting an experimental wireless local area network using diffused infrared communications. Shortly thereafter, in 1980, P. Ferrert reported on an experimental application of a single code spread spectrum radio for wireless terminal communications in the IEEE National Telecommunications Conference. In 1984, a comparison between infrared and CDMA spread spectrum communications for wireless office information networks was published by Kaveh Pahlavan in IEEE Computer Networking Symposium which appeared later in the IEEE Communication Society Magazine. In May 1985, the efforts of Marcus led the FCC to announce experimental ISM bands for commercial application of spread spectrum technology. Later on, M. Kavehrad reported on an experimental wireless PBX system using code division multiple access. These efforts prompted significant industrial activities in the development of a new generation of wireless local area networks and it updated several old discussions in the portable and mobile radio industry[12].

The first generation of wireless data modems was developed in the early 1980s by amateur radio operators, who commonly referred to this as packet radio. They added a voice band data communication modem, with data rates below 9600-bit/s, to an existing short distance radio system, typically in the two meter amateur band. The second generation of wireless modems was developed immediately after the FCC announcement in the experimental bands for non-military use of the spread spectrum technology. These modems provided data rates on the order of hundreds of kbit/s. The third generation of wireless modem then aimed at compatibility with the existing LANs with data rates on the order of Mbit/s. Several companies developed the third generation products with data rates above 1 Mbit/s and a couple of products had already been announced by the time of the first IEEE Workshop on Wireless LANs."[3]

"The first of the IEEE Workshops on Wireless LAN was held in 1991. At that time early wireless LAN products had just appeared in the market and the IEEE 802.11
committee had just started its activities to develop a standard for wireless LANs. The focus of that first workshop was evaluation of the alternative technologies. By 1996, the technology was relatively mature, a variety of applications had been identified and addressed and technologies that enable these applications were well understood. Chip sets aimed at wireless LAN implementations and applications, a key enabling technology for rapid market growth, were emerging in the market. Wireless LANs were being used in hospitals, stock exchanges, and other in building and campus settings for nomadic access, point-to-point LAN bridges, ad-hoc networking, and even larger applications through internetworking. The IEEE 802.11 standard and variants and alternatives, such as the wireless LAN interoperability forum and the European HiperLAN specification had made rapid progress, and the unlicensed PCS Unlicensed Personal Communications Services and the proposed SUPERNet, later on renamed as U-NII, bands also presented new opportunities. [4]

WLAN hardware was initially so expensive that it was only used as an alternative to cabled LAN in places where cabling was difficult or impossible. Early development included industry-specific solutions and proprietary protocols, but at the end of the 1990s these were replaced by standards, primarily the various versions of IEEE 802.11 (commonly misunderstood as equal to trademark Wi-Fi of Wi-Fi_Alliance). An alternative ATM-like 5 GHz standardized technology, HiperLAN/2, has so far not succeeded in the market, and with the release of the faster 54 Mbit/s 802.11a (5 GHz) and 802.11g (2.4 GHz) standards, almost certainly never will [12].

APPLICATIONS OF WIRELESS TECHNOLOGY

Security systems

Wireless technology may supplement or replace hard wired implementations in security systems for homes or office buildings.

Cellular telephone (phones and modems)

Perhaps the best known example of wireless technology is the cellular telephone and modems. These instruments use radio waves to enable the operator to make phone calls from many locations worldwide. They can be used anywhere that there is a cellular telephone site to house the equipment that is required to transmit and receive the signal that is used to transfer both voice and data to and from these instruments.

Wi-Fi

Wi-Fi is a wireless local area network that enables portable computing devices to connect easily to the Internet. Standardized as IEEE 802.11 a,b,g,n, Wi-Fi approaches speeds of some types of wired Ethernet. Wi-Fi hot spots have been popular over the past few years. Some businesses charge customers a monthly fee for service, while others have begun offering it for free in an effort to increase the sales of their goods. [5]

Wireless energy transfer

Wireless energy transfer is a process whereby electrical energy is transmitted from a power source to an electrical load that does not have a built-in power source, without the use of interconnecting wires.

Computer interface devices

Answering the call of customers frustrated with cord clutter, many manufactures of computer peripherals turned to wireless technology to satisfy their consumer base. Originally these units used bulky, highly limited transceivers to mediate between a computer and a keyboard and mouse, however more recent generations have used small, high quality devices, some even incorporating Bluetooth. These systems have become so ubiquitous that some users have begun complaining about a lack of wired peripherals. [who?] Wireless devices tend to have a slightly slower response time than their wired counterparts, however the gap is decreasing. Initial concerns about the security of wireless keyboards have also been addressed with the maturation of the technology.

Categories of wireless implementations, devices and standards

1. Radio communication system
2. Broadcasting
3. Amateur radio
4. Land Mobile Radio or Professional Mobile Radio: TETRA, P25, OpenSky, EDACS, DMR, dPMR
5. Communication radio
6. Cordless telephony:DECT (Digital Enhanced Cordless Telecommunications)
8. List of emerging technologies
9. Short-range point-to-point communication : Wireless microphones, Remote controls, IrDA, RFID (Radio Frequency Identification), Wireless USB, DSRC (Dedicated Short Range Communications), EnOcean, Near Field Communication
10. Wireless sensor networks: ZigBee, EnOcean; Personal area networks, Bluetooth, TransferJet, Ultra-wideband (UWB from WiMedia Alliance).
11. Wireless networks: Wireless LAN (WLAN), (IEEE 802.11 branded as Wi-Fi and HiperLAN), Wireless
Metropolitan Area Networks (WMAN) and Broadband Fixed Access (BWA) (LMDS, WiMAX, AIDAAS and HiperMAN) Ref[14][4]

WIRELESS NETWORK FAULT ANALYSIS

Fault Diagnosis

Diagnosing faults in a network, may it be wireline or wireless, is a difficult problem because of the interactions between the different network entities and the interactions between faults. Fault management in a wireless network is further complicated by the following factors:

a) Such networks are overly prone to link errors with signal propagation affected by fluctuating environmental conditions. This makes the network topology dynamic and unpredictable. Node mobility further aggravates the problems.

b) The capacity of such networks is generally limited. Scarcity of resources such as bandwidth and battery power puts a tight constraint on the amount of management traffic overhead the network can tolerate.

c) Wireless communication is known to be vulnerable to link attacks. Consequently, such networks are highly susceptible to attacks from malicious parties. The attackers can inject false information to disrupt or interfere with the network management effort.

Network Framework

To address the challenges, we propose a novel fault diagnosis framework that uses on-line trace-driven simulation to detect faults and analyse root causes. Our framework is based on reproducing inside a simulator the events that took place in the network. This framework has the following advantages.

First, it is flexible. Since a simulator is often highly customizable and applies to a large class of networks under different environment with appropriate parameter settings, fault diagnosis built on top of it inherits its flexibility.

Second, the use of a simulator enables us to capture the complicated interactions within the network, and between the network and the environment, as well as among the different faults. Therefore it allows us to systematically diagnose a wide range and combination of faults.

Third, the framework is extensible in that the ability of detecting new faults can be built into the framework by modelling the faults in the simulator independent of the other faults in the system—the interaction between the faults is captured implicitly by the simulator. Fourth, reproducing the network inside a simulator facilitates what-if analysis, which offers predictions for network behaviour if certain changes were to be applied to the network. The simulator can provide accurate quantitative feedback to the administrator on the performance impact of possible corrective measures.

Issues in Framework

For the framework to be effective, we need to address two key issues:

(i) How to accurately reproduce what happened in the network inside a simulator;

(ii) How to build fault diagnosis on top of a simulator to perform root cause analysis.

To address the first issue, we drive an existing network simulator (e.g., Qualnet) with traces obtained from the network being diagnosed. Using real traces removes the dependency on generic theoretical models that may not capture the nuances of the hardware, software and environment of the particular network in question, thus improving the accuracy of the system. We quantify the accuracy of trace-driven simulation through a set of real measurements, and show that for the purpose of fault diagnosis, it is possible to use simulations to reproduce what happened in the real network, after the fact. Further, we address the issue of collecting good quality trace data (in face of measurement errors, nodes supplying false information, and software/hardware errors) by developing a technique to effectively rule out erroneous data from the trace.

Network Management Architecture

Our management architecture consists of two types of software modules.

An agent runs on every router node in the multihop wireless network, gathers information from various protocol layers and from the router’s wireless network card, and reports this information to a management server, called manager. It is the manager that performs the analysis and takes appropriate actions. Management of the network could be centralized by placing the manager on a single node, or distributed by running the manager on a set of node.

The fault diagnosis process is illustrated in Figure 3. The process starts by agents continuously collecting and transmitting their (local) view of the network’s behaviour to the manager(s). Examples of the information sent include traffic statistics, received packet signal strength on various links, and re-transmission counts on each link.

Fault Diagnosis Algorithm

We now describe an algorithm to systematically diagnose root causes for failures and performance problems.
**General approach:** Applying simulations to fault diagnosis enables us to reduce the original diagnosis problem to the problem of searching for a set of faults such that their injection results in an expected performance that matches well with observed performance. More formally, given a network settings, $NS$, our goal is to find $FaultSet$ such that $SimPerf (NS; FaultSet) = ObservedPerf$, where the performance is a function value, which can be quantified using different metrics. It is clear that the search space is high-dimensional due to many combinations of faults. To make the search efficient, we take advantage of the fact that different types of faults often change one or few metrics. For example, random dropping only affects link loss rate, but not the other metrics. Therefore we can use the metrics in which the observed and expected performances have significant difference to guide our search.

**Below we introduce our algorithm**

**Initial diagnosis:** We start by considering a simple case where all faults are of the same type, and the faults do not have strong interactions. We will later extend the algorithm to handle more general cases, where we have multiple types of faults, or faults that interact with each other. For ease of description, we use the following three types of faults as examples: random packet dropping, random noise, MAC misbehavior, but the same methodology can be extended to handle other types of faults once the symptoms of the fault are identified. We use trace-driven simulation, fed with current network settings, to establish the expected performance. We consider large deviations from the expected performance as anomalies, and look for the faults that can explain the discrepancy. In particular, based on the difference between the expected performance and observed performance, we first determine the type of faults using a decision tree.

Due to many factors, simulated performance is unlikely to be identical with the observed performance even in the absence of faults. Therefore we conclude that there are anomalies only when the difference exceeds a threshold. The fault classification scheme takes advantage of the fact that different faults exhibit different behaviors. While their behaviors are not completely non-overlapping (e.g., both noise sources and random dropping increase link loss rates; and lowering CW increases the traffic and hence increases noise caused by interference), we can categorize the faults by checking the differentiating component first. For example, external noise sources increase noise experienced by its neighboring nodes, but do not increase the sending rates of any node, and therefore can be differentiated from MAC misbehavior and random dropping (figure 4).

1) Let $NS$ denote the network settings (i.e., signal strength, traffic statistics, routing table) Let $RealPerf$
Network Fault Diagnosis Flow Diagram.

Figure 4. Network Fault Diagnosis Flow Diagram.

denote the real network performance

2) FaultSet = $f_g$

3) Predict $SimPerf$ by running simulation with input $(NS; FaultSet)$

4) if $|Di®(SimPerf ;RealPerf )| > threshold$ determine the fault type $ft$ using the decision tree for each link or node $i$ if $(Di®(SimPerf (i);RealPerf (i))| > threshold)$ add fault($ft ; i$).

Diagnosing and Analyzing the Wireless Network

Rechecking the connections:

As we know that there are several types of connections that are made or established the wireless networking media. So, the connections made between the routing devices and the wireless networking devices that are used to configure the wireless network should be rechecked before configuring the wireless network. It is the initial step of the wireless network troubleshooting.

Rechecking the Settings:

After checking the physical connections between the routers and other wireless networking devices, now its time to recheck the settings of the routing device and of different type of networking adapters either they are properly installed or not. If not then first install it properly before connecting to the network.

TCP/IP Settings:

Now check and reconfigure the settings of the TCP and IP of your wireless networking device. Every networking device that is able to operate wirelessly has a specific IP address. Reconfigure the IP settings and assign the settings that are accurate or closely matched to the hardware or the device. After entering the IP address use 6eh 'Ping' to verify the settings and also check the working of the access point.

Wireless Specific and Security problems:

If still the user does not connect to the network after the valid entrance of the IP address then there must be some wireless problems specific in nature so, recheck them. After rechecking if the problem is their then change the setting of the security of the wireless device and set 6eh accurate or the closely associated settings.

Ensure the RADIUS:

In this step we should check the RADIUS server and recheck the settings of the RADIUS to carry on the wireless working smoothly. If there is some problem in the connection of the RADIUS we should check out the protocol of the RADIUS that is called as Authentication Protocol.

Wireless Network Troubleshooting: Connectivity

When you have trouble connecting a wireless client (a desktop, laptop, PDA, or phone) to an office network, these step-by-step debugging tips can help.

1. Start by rechecking your physical connections -- a common culprit that is often overlooked. Check your
wireless router's WAN port link to your cable/DSL modem and LAN port links to Ethernet clients. Make sure that WAN and LAN cables are inserted tightly and the status lights are on at both ends. If not:

1. Try swapping Ethernet cables to isolate a damaged cable.
2. Check your router's manual to make sure that you're using the right type of cable -- some WAN uplinks require cross-over cables.
3. If status lights are still off, connect another device like a laptop to the affected WAN or LAN port. If status changes, to device you just replaced may be failing link auto-negotiation. Check port configurations at both ends and reconfigure as needed to match speed and duplex mode.

2. Next, verify that your client's wireless adapter is installed and working properly On a Windows client, select your wireless connection from the Network Connections panel and verify that its status is "Enabled."

If the adapter is not listed, there may be a problem with the associated PC Card slot or USB adapter cable. If removing and reconnecting the adapter does not help, use Device Manager to uninstall / reinstall that adapter. If the adapter is listed but the connection cannot be enabled, use the Properties panel to spot resource conflicts or update the driver.

3. Next, verify that your wireless router's LAN settings are correct. Use your router's admin utility to determine its LAN port IP address and subnet.

Make sure the router's DHCP server is set to assign IPs using a non-overlapping range in the same subnet as the LAN port address.

If your router's DHCP Server is set to filter access by MAC address, add your client's MAC address to that "allowed device" list.

Check your router's Log or Status page to verify that an IP address is indeed assigned to your wireless client whenever it connects.

4. Next, verify your client's TCP/IP settings. Although we describe using Windows to manage wireless connections here, troubleshooting is conceptually similar when using any other connection manager (e.g., Intel, Linksys).

Open the Network Connections panel and select your wireless connection. If the status is still "Disabled," return to step 2.

If status is "Not Connected," use View Available Networks to select your wireless network and click Connect. If your network's name does not appear in Available Networks or you cannot connect to your network, go to step 8 to debug wireless settings.

While attempting to connect, status may change briefly to "Acquiring Network Address," then "Connected." At that point, use Status/Support to determine the client's assigned IP address. If the client's IP is 0.0.0.0 or 169.254.x.x, click Repair. If that problem persists, go to step 8.

Otherwise, if the connected client's IP address is not in your router's LAN subnet, use the Properties / Internet (TCP/IP) panel to reconfigure the connection to get an address automatically and repeat step 4.

5. Once your client has a valid IP address, use "ping" to verify network connectivity. Run a command window from the client's start menu and use it to ping your router's LAN IP address as shown in Figure 5.

If pinging your router repeatedly fails, skip to step 6.
If pinging your router is successful, then ping any other wired or wireless LAN client that you wish to share files or printers with. If that ping fails, then the destination may be using a firewall to block incoming messages.

On Windows XP SP1 or earlier, use the connection's Properties panel to temporarily disable the Internet Connection Firewall.

On Windows XP SP2 or later, use the client PC's control panel to temporarily disable the Windows Firewall.

If no Microsoft firewall was running, check for a third-party personal firewall like ZoneAlarm and temporarily disable it.

After disabling the destination's personal firewall, pings that client again. If ping is now successful, then the firewall you disabled may also be blocking Windows Network protocols. Reconfigure the firewall to permit the traffic you want to exchange between LAN clients. For example, to share files and printers, permit incoming NetBIOS connections from your LAN subnet. Don't forget to re-enable the firewall when finished!

6. If your wireless client still cannot connect, get a valid IP address, or ping your router, it's time to consider wireless-specific problems. The router and client must use compatible 802.11 standards and the same network name (SSID). Use your router's admin utility to view WLAN settings and compare them to your client's wireless connection parameters.

If your router's network name does not appear in the client's Available Networks list, enable "SSID broadcasts" on your router. Alternatively, add the network name to your client's Preferred Connections manually. Be sure to match the router's SSID exactly, including capitalization.

With an 802.11b client, you must use an 11b, g, or n router. If any 11b clients are present, enable b+g protection on your 11g router.

With an 802.11a client, you must use an 11a router. For dual-band products, you may configure both ends to use 11b/g, a, or both.

802.11g or 11n clients can generally use 11g or n routers. However, in mixed 11g/n WLANs, disable any vendor extensions (e.g., turbo mode).

7. If a matched wireless client and router can "hear" each other but still cannot connect or exchange traffic, look for a security mismatch. The client must support the security mode required by the router: Open, WEP, WPA, or WPA2. Unless the WLAN is open (unsecured), the router and client must also have (or dynamically receive) the same keys used to encrypt traffic between them. Compare your router's WLAN security settings to your client connections preferred network properties and attempt to match them.

If your router uses WEP, set the client's encryption to WEP and match the router's authentication type (open or shared). Copy the router's first WEP key to the client, translating from ASCII to hex if needed.

If your router uses WPA-Personal, set the client's authentication to WPA-PSK and match the router's encryption type (TKIP or AES). Use the router's passphrase as the client's network key; capitalization counts!

If your router uses WPA2-Personal, set the client's authentication to WPA2-PSK and use the router's passphrase as the client's network key.

If your router uses WPA or WPA2-Enterprise, set the client's authentication to WPA or WPA2 respectively, match the router's encryption type, and continue 802.1X set-up in step 8.

8. Ensure RADIUS is working. WPA and WPA2-Enterprise log the client into the network and deliver encryption keys using an 802.1X-capable RADIUS server. If you do not already have a RADIUS server, otherwise, try the following:

Re-configure your router and server with a matching RADIUS secret.

Re-configure your RADIUS server to accept requests from your router.

Use ping to verify router-to-RADIUS server network reachability.

Watch router LAN packet counters to verify that RADIUS is being sent.

Use a LAN analyzer (such as Wireshark) between the router and server to decode RADIUS.

On XP SP2 clients, entering "netsh ras set tracing * enabled" to write 802.1X debug messages to the Wzctrace.log file.

9. If RADIUS is working but the client's access requests are rejected, look for an 802.1X Extensible Authentication Protocol (EAP) or user login problem. Your client must support one of the EAP types required by your server and must supply a valid login and password / token / certificate or other kind of credential.

If your server requires EAP-TLS, select "Smart Card or other Certificate" on the client's Network Properties / Authentication panel.

If your server requires PEAP, select "Protected EAP" on that panel.

If your server requires EAP-TTLS, install a third-party 802.1X Supplicant program like Juniper OAC or Cisco SSC on the client.

Make sure that client and server EAP-specific properties match, including server certificate Trusted Root Authority, server domain name (optional), and tunneled authentication method (e.g., EAP-MSCHAPv2, EAP-GTC).

If you are prompted to accept the server's certificate at connect time, examine the certificate carefully, verifying
enables us to inject different types of faults in a controlled scheme. Diagnosis techniques and inconsistency detection manager to change the values and influence the behavior performance. Then we evaluate the effectiveness of our data collection and show its impact on the overall performance. We first quantify the network overhead introduced by interface (GUI) to interact with network administrators and management is cast into performance counters supported on Windows. Performance counters are essentially (name, value) pairs grouped by categories. This framework is easily extensible. Adding to the information being monitored simply involves creating a new category of performance counters and writing a module that updates the performance counter values as the information changes. Performance data related to TCP, UDP, IP, and WRAPI have all been incorporated into the framework with little work. Values in these performance counters are not always read-only. Writable counters offer a way for an authorized manager to change the values and influence the behavior of a node in order to fix problems or initiate experiments remotely. Each manager is also equipped with a graphical user interface (GUI) to interact with network administrators. The GUI allows an administrator to visualize the network as well as to issue management requests.

**Evaluation**

We first quantify the network overhead introduced by data collection and show its impact on the overall performance. Then we evaluate the effectiveness of our diagnosis approaches and inconsistent detection scheme. We use simulations for most of our evaluation, since it enables us to inject different types of faults in a controlled and repeatable manner, and quantify the accuracy of our approach. When fault diagnosis is evaluated in a simulator, we diagnose traces collected from simulation runs that have injected faults. In simulation-based evaluation, we mainly focus on validating the following two important components in our approach: trace-driven simulation (in particular, link-based traffic simulation and route simulation) and fault diagnosis algorithm. Finally we report our experience of applying the approach to a small-scale testbed. Even though the results we could obtain from the test-bed were limited by our inability to inject some types of faults like external noise and MAC misbehavior in a controlled fashion and the lack of transparency in the current generation of drivers for wireless NICs, they demonstrate the feasibility of on-line simulations in a real system.

**Evaluation of Fault Diagnosis through Simulations**

**Diagnosing one or more faults of possibly different Types**

Our general methodology of using simulation to evaluate fault diagnosis is as follows. We artificially inject a set of faults into a network, and obtain a set of faulty traces, which includes network topology, link load and routing updates. We then feed these faulty traces into the fault diagnosis module to infer root causes, and quantify the diagnosis accuracy by comparing the inferred fault set with the fault set originally injected.

We use both grid topologies and random topologies for our evaluation. In a grid topology, only nodes horizontally or vertically adjacent can directly communicate with each other, whereas in random topologies, nodes are randomly placed in a region. To challenge our diagnosis scheme, we put a high load on the network by randomly picking 25 pairs of nodes to send one-way constant bit rate (CBR) traffic at a rate of 1 Mbps. Under this load, the links in the network have significant congestion loss, which makes diagnosis even harder. For example, differentiating loss caused by random packet dropping is more difficult when there is significant congestion loss. Such diagnosis time scale is acceptable for diagnosing long-term performance problems. Moreover, the efficiency can be significantly improved through code optimization.

We use coverage and false positive to quantify the accuracy of fault detection, where coverage represents the fraction of faulty locations that are correctly identified, and false positive is the number of (non-faulty) locations incorrectly identified as faulty divided by the total number of true faults. We consider a fault is correctly identified when both its type and its location are correct. For random dropping and external noise sources, we also compare the inferred faults’ magnitudes with their true magnitudes.
Software Requirement

The software required to run this manager package very well are:

Operating System: Window XP, Windows 7, Windows Vista etc
System Software: Network Manager, Network Fault Simulators.

CONCLUSION

This paper has addressed a fault diagnosis problem on a wireless network using a trace-driven simulation approach defining the node connection topology, and a re-lay time estimate associated to each node. We assumed that our interface with the network is a gateway node that accepts ping commands (control input), and replies with pong responses after a time delay that we can measure (observed output).

To summarize, in this section we evaluate our trace-driven diagnosis approach in both simulation and testbed under a variety of scenarios. Our results show that the approach is effective in detecting and diagnosing faults without imposing much overhead on the network. Moreover we also show that the trace-driven simulation approach can be used for what-if analysis to evaluate the performance of alternate network configurations.

REFERENCES

[7] Linux Aironet driver - comments and source were invaluable - http://sourceforge.net/projects/airo-linux/