Diagnosing Failures in Wireless Networks using Fault Signatures

Dhruv Gupta
Dept. of Computer Science
University of California Davis
Email: dhgupta@ucdavis.edu

Prasant Mohapatra
Dept. of Computer Science
University of California Davis
Email: prasant@cs.ucdavis.edu

Chen-Nee Chuah
Dept. of Electrical Engineering
University of California Davis
Email: chuah@ucdavis.edu

***Abstract*—Detection and diagnosis of failures in wireless networks is of crucial importance. It is also a very challenging task, given the myriad of problems that plague present day wireless networks. A host of issues such as software bugs, hardware failures, and environmental factors, can cause performance degradations in wireless networks. As part of this study, we propose a new approach for diagnosing performance degradations in wireless networks, based on the concept of "fault signatures". Our goal is to construct signatures for known faults in wireless networks and utilize these to identify particular faults. Via preliminary experiments, we show how these signatures can be generated and how they can help us in diagnosing network faults and distinguishing them from legitimate network events. Unlike most previous approaches, our scheme allows us to identify the root cause of the fault by capturing the state of the network parameters during the occurrence of the fault.**

I. Introduction

Wireless networks have become increasingly prevalent over the last few years and are being widely deployed across universities and enterprises ([1] [2]). Administrators of these networks have to deal with a variety of changes such as variation in topology, changing user behavior, and others, that can impact network performance. The fact that communication networks have become critical in today's world calls for an efficient mechanism that can help in the smooth functioning of the network. In the case of wireless networks, this problem is even more acute as environmental factors also become a critical component along with the above mentioned issues. It is extremely important to be able to detect and diagnose problems correctly, and quickly, in order to minimize their impact on the end users' performance.

Several tools exist in order to troubleshoot present day wireless networks. However, a major drawback of the existing tools is that they are unable to distinguish between the root causes of various performance degradations. This happens because these tools tend to consider higher layer parameters such as traffic load, delay, and packet loss as indicators of network health. These metrics tend to aggregate the variations of multiple MAC and physical layer parameters such as retransmissions, signal strength, noise floor, and modulation rate. As a result, even though a performance degradation is detected, the cause of the fault is often misdiagnosed, or in some cases not even identified.

In this work, we propose the idea of constructing signatures for commonly occurring faults in wireless networks. The basic premise of our work is that these faults will manifest themselves as performance degradations. Our goal is to model these performance degradations as network anomalies and use a signature-based anomaly detection scheme for identifying network faults. We propose to build a statistical model of "normal" network performance and identify network anomalies as events where network performance deviates from this normal behavior. We also propose to construct unique "fault signatures" for commonly occurring faults in wireless networks. In particular, by analyzing individual faults, we can learn how these faults result in anomalous behavior for certain network parameters, while not impacting other parameters. This characterization of network faults can potentially be used to distinguish one fault from another. By comparing the current network state against the fault signatures, we can detect the presence of these faults in the network and analyze their root cause. The key contributions of our work are:

- **As our first contribution, we show how existing network diagnosis tools may misdiagnose performance degradations in wireless networks, or even confuse them with legitimate network events.** We further identify different network parameters from multiple protocol layers and show why it is necessary to follow a cross-layer approach to address this problem.
- **As our second contribution, we explore the idea of constructing signatures for some commonly occurring faults in wireless networks.** We simulate these faults by purposefully injecting them into our laboratory testbed and collect statistics both during the normal functioning of the network and when the faults are introduced. By capturing the state of the network parameters during the occurrence of the faults, we are able to build unique signatures that can help us distinguish one fault from another.
- **As our third contribution, we introduce a scheme for network diagnosis, based on the concept of fault signatures**. We show how we can statistically model the network's performance as normal and anomalous. By abstracting out

the state of the network parameters, we create a template for the normal behavior of the network, which will capture the time-varying nature of the network. By matching this performance template against the fault signatures, we can detect the occurrence of a network fault, diagnose its cause, and differentiate it from legitimate changes in the network environment.

Paper Outline. Section II outlines some of the previous work, along with the motivation behind our work. In Section III we describe how to abstract the normal behavior of the network and classify data points as normal or anomalous. Section IV outlines the process of generating fault signatures. In Section V, we conclude the paper by discussing the challenges and future work.

II. Related Work & Motivation

Detection and diagnosis of faults in wireless networks remains an open issue. Previously proposed solutions for troubleshooting wireless networks can be primarily classified into two categories. The first one involves using active measurements for detecting faulty links ([3] [4] [5]). However, such schemes focus only on detecting lossy links, and not performance degradations in general. Moreover, they introduce extra overheads in the wireless network. The second class of works are based on the idea of using passive wireless sniffers. Traces from several sniffers are merged together to build a complete picture of the network that is used for network troubleshooting ([6] [7] [8]). The problem with such approaches is that they require dedicated monitoring agents to be deployed in the network and only provide offline troubleshooting. Moreover, the problem of collecting and merging traces from several different sniffers is not trivial and requires tight synchronization among the sniffers. There has also been some previous work on generating signatures for IEEE 802.11 networks. However, these works have either focussed on fingerprinting wireless radios and device drivers [9], or on how to uniquely identify a wireless network user [10].

In the area of signature-based anomaly detection, several previous works have focussed on wireless networks. These works have used both machine-learning based and time-series analysis based techniques for anomaly detection ([11] [12] [13] [14]). However, the primary focus of these works has been on detecting security attacks (such as DDoS and wormhole attacks), and not performance degradations and failures.

The work that is closest to ours appears in [15]. In this work, the authors have proposed analyzing physical layer parameters such as signal strength and noise floor in order to differentiate between faults such as hidden terminal and packet capture. However, the authors focus only on the physical layer parameters (and subsequently only on faults that can be detected by these parameters) and do not utilize the information available at the link layer. Variations in physical layer parameters do not necessarily reflect the state of the link layer parameters, as contended by the authors. Our work focusses on a far broader range of performance degradations, while utilizing information from across the protocol stack. Another related piece of work appears in [16]. Here, the authors propose using fault feature vectors for detecting specific faults in Ethernet networks, while our goal is to develop signatures for faults that occur in wireless networks.

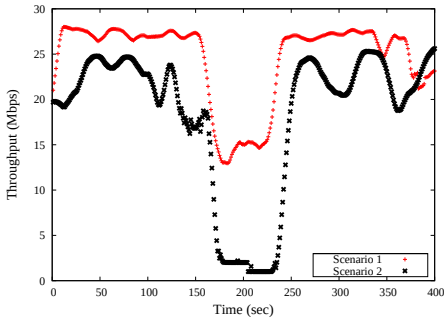

Fig. 1. Impact of two different scenarios on user throughput. Scenario 2 corresponds to an actual fault, while scenario 1 simply involves a new client being added into the network.

A. Fault Misdiagnosis

A major short coming of most previous approaches is that they are unable to diagnose the root cause of the network failure. In several instances, these tools may even misdiagnose legitimate network events as performance degradations. *Legitimate changes in the network (changing user profiles, launching of new applications and others) can also cause the network to depart from its normal behavior, thereby making it appear as if a fault has occurred.* In order to illustrate this problem, consider the two plots in Figure 1. These plots show a snapshot of our wireless network while a file transfer operation was being performed between two nodes. During the time period from 150 to 250 seconds, we can see how the sender's throughput is adversely impacted for the two different scenarios. While scenario 2 in the figure represents a genuine fault occurring in the network (we moved the destination node farther away from the source node resulting in reduced link quality), the first scenario only involves a new client being introduced in the network for that time period (a legitimate network event). Hence, in order to be able to distinguish between such scenarios, we need to consider a variety of network metrics from different protocol layers.

III. Methodology

Our basic premise is that network faults will result in performance degradation and will cause the network to deviate from its normal behavior. This deviation in the network's behavior can be used to identify potential anomalies. However, we also need to take into account the time varying nature of the network status. Network usage will vary as user profiles change over time and network components are added or removed. Hence, in order to be able to correctly diagnose the network, it is crucial that our fault detection system

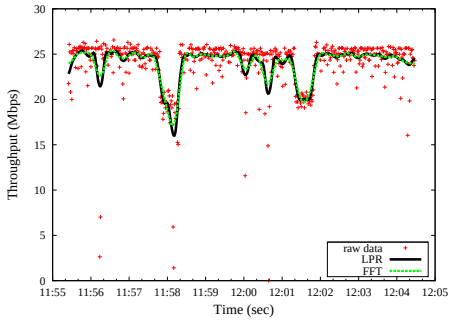

Fig. 2. The scatter plot shows the raw throughput data for a particular link in the wireless network. The line plots show the smoothed data that can serve as the template for normal network behavior.

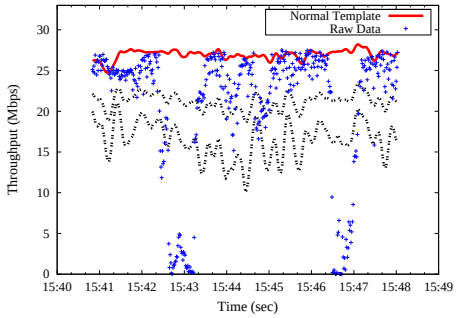

Fig. 3. The line plots show a snapshot of the normal performance template for network load along with the lower 3rd and 6th standard deviations. The scatter plot shows the raw data for the same time period for a different day.

adapts itself to these changes in the network environment. Our scheme involves learning the normal behavior of the network via continued observation, and classifying future events and observations as normal or anomalous based on past experiences.

A. Smoothing Raw Data

The scatter plot in Figure 2 shows the raw throughput data for one link on our testbed. As can be seen, there is considerable variation over time, along with several outliers. For the purpose of generating a template to represent the "normal" behavior of the network, we need to capture the general trend or pattern in the data, while diminishing the impact of outliers. At the same time, we need to avoid over-smoothing the data and missing significant fluctuations. In order to achieve this, we evaluated two different techniques. In *LPR*, we pass our data through multiple steps of smoothing, involving adjacent averaging and local polynomial regression. In *FFT*, we employ a FFT-based filter to smooth the data. Both techniques gave fairly similar results, and we use the *LPR* technique for rest of the evaluation.

Once the data has been smoothed, it can be integrated into the current network template by using techniques such as exponential smoothing. Exponential smoothing can help us incorporate the effects of persistent events into the template, while diminishing the impact of temporary variations in network performance. The rate at which the template adapts itself to change will be determined by the smoothing constant, whose value will depend upon the target network and how stable or dynamic the network is.

B. Fault Detection

We propose a fault detection system wherein the network's performance is classified into "normal" and "anomalous" by performing statistical analysis. Moreover, if the performance is deemed anomalous, we can classify as to how anomalous it is. In order to achieve this, we first need to define performance thresholds which will decide whether a data point is normal or anomalous. Currently, we use the third and the sixth standard deviations as our performance thresholds, as they have been commonly used in the area of statistical quality control in the past ([17]). The performance is deemed as normal (and assigned a score of '0') if it is within the upper and lower third standard deviations. All data points between the third and the sixth standard deviations are given a score of 1 or -1 depending on whether they are above or below the average. Similarly, all data points beyond the sixth standard deviation are given a score of '2' or '-2' (above or below the average). At any given instant of time, the state of the network parameters would constitute what we call the *performance template* of the network, which will help us to represent the network's behavior in a compact manner.

To highlight this with an example, consider the plot in Figure 3. Here we show a snapshot of the normal network behavior in terms of network load on a particular link. The line plots show the average load along with the lower third and sixth standard deviations (the upper thresholds are not shown for clarity). The scatter plot shows the raw load data on the same link, for the same time period on a different day. Our fault detection scheme would mark the points that fall below or above the performance thresholds as anomalous and assign them the respective scores. All the points within the acceptable range would be deemed as normal and get a value of '0'. A series of digressions outside the threshold can indicate a potential fault, thereby helping us in identifying any potential performance degradations in the network.

IV. Generating Fault Signatures

The main idea behind our work is to construct signatures for various network faults and use them to identify when that particular fault occurs in the network. In order to construct these signatures, we use our laboratory testbed and purposefully inject these faults into the network. We collect data traces and then analyze them in order to observe any patterns that might be visible. Once we can construct individual signatures for various faults, we can match them against the network performance template and use it for fault detection.

A. Evaluation Methodology

We generated signatures for the following four scenarios:

- The first scenario is termed as **Low Sig**, denoting low signal quality. This represents the case where receiver mobility

results in reduced signal strength at the transmitter, thereby causing performance degradation. We simulate this fault by moving the transmitter away from the receiver.

• The second scenario is another fault which we call **Low Power**. In this case, we assume that the transmitting node uses lower power (for power conservation or any other reason), resulting in reduced link quality. We simulate this fault by lowering the transmission power of the sender's radio during data transfer.

• The third scenario is called **BSTRM** and stands for Broadcast Storm. In this scenario, one of the nodes floods the wireless network with broadcast ping messages resulting in degraded network performance.

• The fourth scenario is called **Int**, denoting interference. This is an example of a legitimate scenario wherein a new client node joins the network. Due to a shared transmission medium, this will decrease the throughput of the previously existing node and can be mistaken as a potential network fault.

The first step towards generating fault signatures is the selection of parameters that we intend to use for the signatures. The number of parameters should neither be too small, resulting in inaccurate fault signatures and missed faults, nor should we include irrelevant parameters into the fault signature. However, as was shown earlier, we need to consider parameters from across the protocol stack in order to accurately diagnose the root cause of the failure. We selected six metrics and studied their variation over time. These metrics are throughput, inter-arrival delay, frame loss, packet count, MAC-layer re-transmissions, and received signal strength (RSSI). The goal is to define the normal and anomalous behavior of these metrics, both during the presence and absence of faults in the network.

Our testbed consists of eight Linux-based devices deployed on a single floor of a building, running the Madwifi wireless driver. We use the *tcpdump* tool to capture network layer traces of the network's performance. These traces help us infer metrics such as throughput and inter-arrival delay. We further use a network monitoring tool described in [18] to capture various physical and link layer metrics such as RSSI, transmission rate, and link layer re-transmissions. We purposefully inject the above mentioned faults in our testbed during the normal functioning of our network in order to capture the required data.

B. Signature Generation

As mentioned previously, the fault signature is an indication of the network's status during the occurrence of that fault. We capture the state of various performance related parameters during the time when the fault occurs in the network and identify which parameters manifest themselves in an anomalous fashion. This representation of the anomalous state of each performance parameter in the presence of the fault defines that fault's signature. The signature will consist of n integers corresponding to the normal or anomalous state of the n chosen network parameters. In our current implementation, each parameter can take six values: '0' corresponding to normal and '2', '1', '-1' or '-2' corresponding to anomalous (explained in the previous section). A score of X means that the state of this parameter is irrelevant to the fault.

Time	Thr.	Delay	Loss	Pkt. Count	Rx RSSI	No. of Re-Tx
15:41:26	[-2	2	2	-2	-2	1]
15:41:27	[-2	2	2	-2	-2	1]
15:41:28	[-2	2	2	-2	-2	2]
15:41:29	[-2	2	2	-2	-2	2]
15:41:30	[-2	2	2	-2	-2	2]
15:41:31	[-2	2	2	-2	-2	2]
15:41:32	[-2	1	2	-2	-2	2]
15:41:33	[-1	2	2	-1	-2	2]
15:41:34	[-2	2	1	-2	-2	1]
15:41:35	[-2	2	1	-2	-2	2]
15:41:36	[-1	2	2	-1	-2	2]
15:41:37	[-2	2	2	-2	-2	2]
15:41:38	[-2	1	2	-2	-2	2]
15:41:39	[-2	2	2	-2	-2	2]
15:41:40	[-1	2	2	-1	-2	2]
Fault Signature	[-2	2	2	-2	-2	2]

TABLE I
GENERATING THE FAULT SIGNATURE FOR "LOW SIG". THR. REFERS TO THROUGHPUT, RX RSSI REFERS TO RECEIVED SIGNAL STRENGTH, AND RE-TX REFERS TO NUMBER OF LAYER 2 RE-TRANSMISSIONS.

We will use the "Low Sig" fault to further explain how a fault signature can be generated. Table I shows how the fault signature for "Low Sig" can be generated using a fifteen second snapshot of the network during which time the fault was introduced in the network. We observed that while some metrics assumed constant values over the time window, the values of other parameters varied with time. We require specific rules in order to account for such variation in parameter values. One such rule we use is the majority rule, wherein if the parameter retains a particular value for a major portion of the time window (for example 80% or more), that majority value is used as the final value. Based on this rule, we extract the fault signature for "Low Sig" as shown in the table. Another example of such a rule could be that if a parameter retains values with the same sign but different magnitude (for example -1 and -2) within the time window (with no clear majority), we can assign the lower magnitude value (-1 in this case). We can also assign the value X in case none of the rules can be applied, which will indicate that the value of this parameter is unrelated to the fault. For example if the parameter takes all three values (1, 0, and -1) within the time window (with no clear majority), then none of the above mentioned rules can be applied. We utilized repeated occurrences of the same fault to optimize the corresponding signature. For example, we simulated the "Int"

Scenario	Fault Signature
Low Sig	[-2 2 2 -2 -2 2]
Low Power	[-2 2 2 -2 X 2]
Int	[-2 2 2 -2 X X]
BSTRM	[-2 2 2 0 X X]

TABLE II
FINAL FAULT SIGNATURES FOR THE FOUR SCENARIOS. THESE SIGNATURES WERE OPTIMIZED OVER MULTIPLE SIMULATIONS OF THE CHOSEN SCENARIOS.

scenario multiple times on our testbed and noticed that the "No. of Re-tx" parameter did not assume any single value for a major portion of the time window for all repetitions. Based on these observations, we concluded that this particular parameter does not have any relevance for the "Int" scenario and should be assigned a value of X in the final fault signature. Table II shows the final fault signatures for our four chosen scenarios. We were able to generate unique signatures for the selected faults using the chosen set of parameters. These signatures can be matched against the network performance template (Section III-A) in order to diagnose network faults. Similar signatures can be generated for a wide variety of faults such as hidden terminals, external noise, and so on. The parameter set can also be extended to include more metrics such as modulation rate, noise floor and others.

V. CONCLUSION & FUTURE WORK

In this work, we propose a novel technique for diagnosing faults in wireless networks. We propose to model performance degradations as network anomalies and use a signature-based scheme for differentiating between various faults. We propose the concept of "fault signatures" wherein we represent a fault in terms of the anomalous state of various network parameters. By capturing the variations of various network metrics during the occurrence of these faults, we can define a unique signature for each fault. Initial results from our testbed suggest that we can use this approach to distinguish between various faults and can potentially utilize it for network diagnosis. Using signatures will enable us to diagnose the network quickly, and correctly, while maintaining low false positives. A significant advantage of our proposed approach over previously existing works is that it will help identify the root cause of the performance degradation, instead of just classifying it at a higher level. Our proposed approach also involves several challenges that we plan to address in our future work:

• The performance template for normal network behavior needs to evolve with time and adapt itself to changes due to persistent events, while discarding the impact of transient events.

• A systematic approach is required to decide which parameters should be included in the process of generating fault signatures. How to optimize these signatures over repeated occurrences of the fault is also an important problem.

• The problem of specifying rules that can help decide whether a parameter is anomalous during the presence of a fault, and deciding performance thresholds, also requires further study.

As part of our future work, we plan to build a performance anomaly detection system that utilizes the concept of fault signatures to detect performance degradations in wireless networks and evaluate it on our wireless testbed.

REFERENCES

[1] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "Design considerations for a multihop wireless network testbed," *Communications Magazine, IEEE*, vol. 43, 2005.

[2] D. Wu, D. Gupta, S. Liese, and P. Mohapatra, "Qurinet: quail ridge natural reserve wireless mesh network," in *the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2006.

[3] K. Naidu, D. Panigrahi, and R. Rastogi, "Detecting anomalies using end-to-end path measurements," in *27th IEEE International Conference on Computer Communications (INFOCOM)*, April 2008.

[4] H. X. Nguyen and P. Thiran, "Using end-to-end data to infer lossy links in sensor networks," in *25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[5] B. Wang, W. Wei, W. Zeng, and K. Pattipati, "Fault localization using passive end-to-end measurement and sequential testing for wireless sensor networks," in *6th Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON)*, 2009.

[6] Y. C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: solving the puzzle of enterprise 802.11 analysis," in *SIGCOMM Computer Communications Review*, 2006.

[7] S. Nanda and D. Kotz, "Mesh-mon: A multi-radio mesh monitoring and management system," in *Computer Communications*, 2008.

[8] L. Qiu, P. Bahl, A. Rao, and L. Zhou, "Troubleshooting wireless mesh networks," in *Computer Communications Review*, 2006.

[9] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *15th USENIX Security Symposium*, 2006.

[10] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom)*, 2007.

[11] N. Samaan and A. Karmouch, "Network anomaly diagnosis via statistical analysis and evidential reasoning," *Network and Service Management, IEEE Transactions on*, 2008.

[12] A. Ward, P. Glynn, and K. Richardson, "Internet service performance failure detection," *SIGMETRICS Performance Evaluation Review*, 1998.

[13] M. V. Mahoney and P. K. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.

[14] M. Natu and A. S. Seth, "Using temporal correlation for fault localization in dynamically changing networks," *International Journal of Network Management*, 2008.

[15] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker, "Mojo: a distributed physical layer anomaly detection system for 802.11 wlans," in *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, 2006.

[16] F. Feather, D. Siewiorek, and R. Maxion, "Fault detection in an ethernet network using anomaly signature matching," in *Conference on Communications architectures, protocols and applications (SIGCOMM)*, 1993.

[17] H. M. Wadsworth, K. Stephens, and A. Godfrey, *Modern Methods for Quality Control and Improvement*. John Wiley and Sons, 1986.

[18] D. Wu, P. Djukic, and P. Mohapatra, "Determining 802.11 link quality with passive measurements," in *IEEE International Symposium on Wireless Communication Systems*, 2008.