Antler: A Multi-Tiered Approach to Automated Wireless Network Management

Ramya Raghavendra, Prashanth Aravinda Kumar Acharya, Elizabeth M. Belding, Kevin C. Almeroth
Department of Computer Science, University of California, Santa Barbara CA 93106
{ramya, acharya, ebelding, almeroth}@cs.ucsb.edu

Abstract—**Management of a large scale wireless network, be it an infrastructured WLAN or a metro-scale mesh network, presents several challenges. Troubleshooting problems related to wireless access in these networks requires a comprehensive set of metrics and network monitoring data. Current solutions gather large amounts of data and require significant bandwidth and processing to offload and analyze this management traffic. As a result, these solutions are typically not scalable or real-time. To this end, we propose a multi-tiered approach to wireless network monitoring that dynamically controls the granularity of data collection based on observed events in the network. Our approach can achieve significant bandwidth savings and enable real-time automated management of a wireless network. Our initial analysis using traces from a large WLAN shows a significant reduction in the amount of data collected to diagnose problems in a WLAN.**

I. Introduction

Large scale wireless networks in the form of campus-wide infrastructure WLANs and metro-scale IEEE 802.11 mesh networks have proliferated to become an important method of providing Internet connectivity. These networks consist of hundreds to thousands of APs (or mesh routers in metro-scale mesh networks) and are used by thousands of users. The management and troubleshooting of these large wireless networks present several new challenges compared to traditional Ethernet-based wired networks.

One of the factors that contributes to the challenges of effective management of wireless networks is that the performance of the devices in these wireless networks may be impacted by entities outside the network, i.e. the surrounding environment or devices that are not part of the network but share the frequency spectrum. In addition, the large number of proprietary protocols and algorithms used by different IEEE 802.11 client vendors and the interaction among these clients is not well understood. Finally, unlike wired networks, the physical location of the devices provides a strong spatial aspect to all data used in management and troubleshooting of wireless networks.

Due to the inherent uncertainty in the wireless medium, network administrators require a comprehensive set of data and metrics to deal with problems in 802.11 networks. The data include metrics from the 802.11 MAC layer and the PHY layer, in addition to those from higher layers of the stack. Most commercial WLANs use a small fraction of these metrics in order to minimize the data collection and processing overhead. However, previous research has shown that the diagnosis and root cause analysis of many network faults requires a complete trace of the packets in the network [1], [2]. Unfortunately, the capture and analysis of all data packets is not scalable. It significantly increases the bandwidth utilization of the wired backbone connections in a WLAN. Such an approach is infeasible in a mesh network where a majority of the routers do not have a wired backhaul link to transmit the packet capture traces. Even in situations where the wired connection is able to meet the high bandwidth requirements, the packet trace approach has other problems. The processing of the packet traces is a resource intensive computational task and may be unsuitable for real-time remediation of network problems. From our own experience in the development of a real-time network visualization tool, we found that the speed of metric collection/generation, rather than the visual rendering of the data, is the computational bottleneck [3].

For the above reasons, there is a need for a new methodology of metric collection in a wireless network. This new methodology should be bandwidth-efficient, scalable with respect to the number of devices in the network, and at the same time provide a comprehensive set of metrics that can be used to identify all problems/anomalies in the network. Such a methodology would facilitate centralized administration of a large network and also enable the use of tools, such as network visualization, to monitor the network health in real-time.

We propose an adaptive wireless network monitoring solution called Antler. The principal feature of Antler is dynamic and scalable hierarchical collection of metrics in the wireless network, which is an essential first step towards effective network management and troubleshooting. A key observation that guides the design of Antler is that comprehensive metric collection is required only when there are problems in the network. A small subset of these metrics are sufficient when the network performance is satisfactory, and can be used for coarse identification of potential problems. We propose a stateful method that intelligently adapts the metric collection process to capture the most relevant set of metrics. The goal of our system is to reduce the volume of data that needs to be collected and processed without sacrificing the ability to diagnose problems in the network.

A brief overview of the operation of Antler is as follows. The baseline operation consists of collection of a minimal set of metrics that indicate the health of the network. These metrics are constantly monitored and compared against pre-determined thresholds or triggers. When the baseline metrics indicate the possible presence of a problem, the system transitions to the next tier of data collection, which consists of a more detailed

978-1-4244-2219-7/08/$25.00 (c)2008 IEEE

set of metrics. If the second level metrics indicate a problem that requires deeper investigation, the system transitions to the third tier of metric collection, and so on. Alternately, if the problem can be detected and solved using the second level metrics, the system eventually returns to the baseline operation of collecting first tier metrics.

II. Related Work

Network management, health monitoring and fault diagnosis in WLANs has been an area of active research in recent years.

MOJO [4] is an 802.11 troubleshooting system that outlines the importance of detailed physical layer metrics for problem diagnosis and demonstrates that many higher layer symptoms are manifestations of problems at the PHY layer. Adya et al. propose modifications to all clients in the network to assist in troubleshooting [5]. APs in the network act as "software" sensors by capturing wireless metrics and thereby avoid the cost of deploying special sensors. The authors also propose that clients associated with APs can act as a conduit for diagnostic traffic from clients not associated with APs. WiFiProfiler [6] uses a similar client conduit approach but troubleshooting can be done in a peer-peer fashion. However, given the heterogeneous client devices, instrumenting all of them may not be possible. Our solution does not assume any assistance from client devices.

Jigsaw [1] is a comprehensive fault diagnosis system that uses a large set of dedicated wireless radio monitors to observe and record every transmission in a WLAN. The radio monitors send the captured packet trace to a central repository where the packet traces are merged to produce a single time-synchronized trace that provides a detailed view of the sequence of events. The Jigsaw system was later extended to provide automated cross-layered diagnosis of problems [2]. Although Jigsaw provides a complete view of the events in the network, it requires high overhead in terms of infrastructure. The dedicated wireless radio monitors require a backhaul network connection that consumes roughly five times the actual network traffic [2]. The high bandwidth requirements for Jigsaw make it unsuitable for a multi-hop mesh network. Additionally, the scaling properties of the trace merging process in a larger or heavily-used network are not clear.

In addition to work from the research community, there are several commercial tools that are designed for this purpose [7], [8], [9]. The proprietary nature of these tools restricts the available information to feature-sets. Based on the available documentation, we hypothesize that some tools use high level metrics accessed through SNMP MIBS [7] and other tools such as AirMagnet [9] use special radio monitors deployed throughout the network to collect packet traces.

III. Design of Antler

In this section we first present the network architecture in which Antler operates. We then present a brief overview of the design philosophy, followed by a detailed description of the design.

A. Network Model

Our proposed solution is designed for an infrastructured WLAN network. All APs in the WLAN run Antler and communicate with a central controller. The central controller performs the following functions: it collects data from the APs and stores the data in a database; it issues commands to APs to control the data collection; and it provides data to the network administrator. The data collected by the controller may also be accessed by a network health monitoring tool such as SCUBA [3]. In the future, the collected data can also be used for automatic rule-based remediation of problems.

We assume the client devices to be autonomous and largely outside the control of the network administrator. Currently, we restrict the focus of the metric collection system to the wireless access part of the network and do not consider metrics from high layers (e.g. events from DHCP, DNS queries). A majority of the metrics used by our system are supported by several commercial APs and can be accessed through SNMP MIBs [7]. We require minimal modification to the APs to collect new metrics required for our system. In the future we intend to incorporate statistics collected from cooperative client devices, i.e. devices that can communicate with the Antler controller.

B. Design Philosophy

The basic idea in the design of Antler is to use a few baseline metrics that capture the general health of the network. When problems are detected, the system intelligently increases metric collection to capture only those metrics that are needed to diagnose the root cause of the problem. The principle behind the design of such a system is that in the general case networks are in a stable state, during which time it is sufficient to have a light-weight monitoring system. On the other hand, when a problem arises, collection of detailed packet level traces in the area where the problem is detected can facilitate fine-tuned problem diagnosis.

In the design of Antler, we use the concept of tiers of metrics, wherein each tier collects a level of detail more than the previous level. The system goal is to diagnose the network problem at the lowest possible tier, i.e. with the minimum level of detail necessary. When diagnosis cannot be made with certainty at a particular tier, the next tier is triggered to collect more metrics. The biggest challenge in designing a multi-tiered metric collection system is to identify the metrics that are necessary and sufficient for making decisions at each tier for the particular problem set that the system should handle. The classification of metrics into tiers is presented in Section III-E.

One design consideration is whether to make the monitoring system centralized or distributed. The intelligence to transition the metric collection among the different tiers can be either at the central controller or at the APs. The latter option provides a distributed approach that may scale better as the size of the network increases. The metric collection process would also be more responsive to local events. Further, a distributed approach would reduce all monitoring traffic. On the other hand, the central controller has a global view of the network and may be able to correlate symptoms of nearby APs, which we plan to

978-1-4244-2219-7/08/$25.00 (c)2008 IEEE

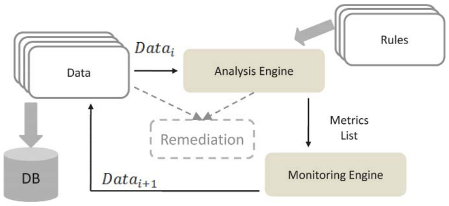

Fig. 1. Architecture of Antler.

explore in an extended version of this work. Additionally, the multi-tiered metric collection scheme reduces the amount of data to process and thus, if done carefully, a central controller will not be overwhelmed with data. Therefore, we opt for the centralized approach to take advantage of the central controller's ability of see a global view of the network.

Next we describe the system architecture of Antler, followed by the classification of metrics into tiers and the rules/triggers that govern the transition of the metric collection among the different tiers.

C. Antler Architecture

Figure 1 illustrates the architecture of the Antler system at the central controller. As seen from the figure, the system is comprised of two primary components - the monitoring engine and the analysis engine. The monitoring engine takes as input a list of metrics to collect at each AP. The list of metrics to collect depends on the current metric collection tier for the particular AP and may differ among the various APs in the network. The monitoring engine interfaces with the APs to collect the corresponding metrics (not shown in figure). The collected metrics are output to the analysis engine and also stored at a central database.

The analysis engine is responsible for generating the list of metrics to collect at each AP. For this purpose, the data collected via the monitoring engine is processed using a repository of rules. The rules are specifications of network conditions or events that trigger a transition in the metric collection tiers. Each rule represents the possible presence of a problem in the network and is usually described in terms of conditional statements that compare current metric values against pre-determined threshold values. Since each rule is inherently associated with a tier of metric collection, the set of rules forms a structure similar to a decision tree. Section III-E provides a detailed discussion of the rules and the decision tree. A second output of the analysis engine is the diagnosis of faults in the network. Based on the problem hypothesis presented by the rules and the corresponding metrics, the analysis engine can perform root cause analysis in the network and suggest potential remedial actions.

D. Metric Selection

There are multiple metrics that can be used to understand the health of a wireless network: throughput, airtime, control overhead, loss rates, retransmissions, data rate, and received signal strength are all good candidates. When a change occurs in the network condition, it is often reflected in one or more of these metrics.

To select a baseline set of metrics, we consider the typical goals for deployment of WLANs [10]. Broadly, there are two goals that the network tries to achieve: 1) provide connectivity to clients within the network's coverage area, and 2) ensure a minimum throughput to all the connected clients (up to the number of clients that the network is designed to support). These two metrics lead us to two of the baseline metrics: maximum client Overhead Index (O_{max}) and minimum client throughput (T_{min}). The Overhead Index is defined as the ratio of control and management traffic to data traffic (in bytes) [11]. When a client has connectivity problems, O_{max} will be high. The second baseline metric, T_{min}, tracks the performance of connected clients. When a client obtains low throughput, T_{min} will be low.[1] In addition to the above two metrics, we also collect the airtime metric (A) for each AP. Airtime, also called channel utilization, is the fraction of time for which the channel is busy and represents the degree of network activity in the neighborhood of the AP [12]. As we show later in the paper, this metric provides valuable supplementary information that helps the decision tree of the analysis engine. If one of the objectives of the network is to support Voice-over-IP applications, then low packet delays is another goal of the network. In such networks, packet delay would be the fourth baseline metric. In this paper we consider WLANs that are unaware of specific traffic type.

We believe that these metrics are sufficient at the high level to detect a network problem. Three of the most important problems related to wireless network access are connectivity problems, performance problems, and authentication problems [5]. Connectivity and authentication problems result in high O_{max}, whereas performance problems result in a low T_{min}. While performance problems manifest in a variety of other metrics, such as round trip time (RTT), data rates, and signal strength, these metrics can be used to provide a deeper understanding for the cause of low throughput and thus are not first tier metrics.

E. Decision Tree

The analysis engine (AE) of Antler, along with the set of rules that specify network conditions, forms a decision tree. Each rule has two parts. The first part is the trigger, which checks for a particular hypothesis of problematic network condition. The trigger is expressed in terms of a combination of the collected metrics compared against predefined thresholds (e.g. $T_{min} < T_{threshold}$). When triggered, the system transitions to the next tier of metric collection to collect more data for finer problem diagnosis. The second part of a rule is the list of metrics to collect in this next tier. Note that a rule in the decision tree at tier n includes all the rules along the path in the tree at tiers $n-1, n-2, \ldots, 1$. Figure 2 provides a graphical representation of the decision tree. We explain the process of choosing thresholds later in this section.

[1] In order to distinguish clients with little or no offered load, we only consider active clients (defined by a minimum activity threshold in bytes transferred) for computation of T_{min}.

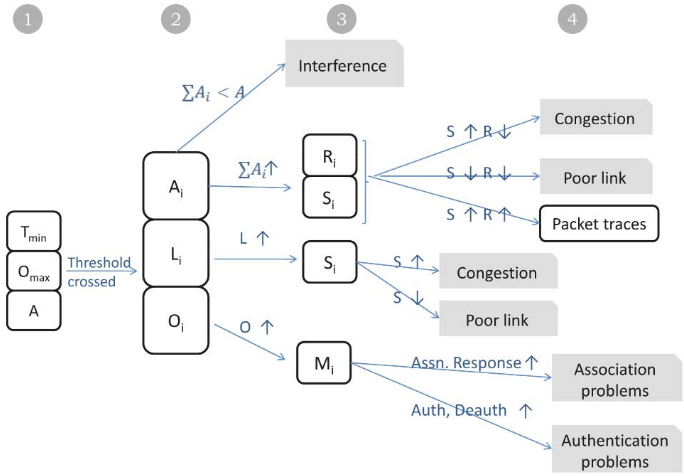

Fig. 2. Multi-tiered metric collection decision tree implemented in the analysis engine. The numbers in circles at the top indicate the tier of metric collection. White boxes represent the metrics collected at each tier. Arrows indicate the triggers used to transition between tiers. Gray boxes indicate the fault diagnosis.

At the first tier of collection, each AP reports the minimum throughput (T_{min}), maximum overhead (O_{max}) and overall airtime (A). During each time interval, the AE compares the reported values against thresholds defined as a part of the rules. At this tier, we ensure that all connected clients are obtaining a minimum throughput and not experiencing connectivity issues, and the network is not nearing the congestion point. To facilitate this, there is a rule to check each of these metrics. The throughput check fails if the throughput of any active client is lower than the threshold, i.e. the minimum throughput specified by the network goal. The overhead check fails if any client (not necessarily active) has an overhead index higher than 100. Finally, the airtime check fails if the overall channel utilization reported by an AP is above 60% [12]. If any of these thresholds are exceeded, the second tier of metric collection is initiated.

In the second tier of metric collection, we obtain additional metrics that help troubleshoot the cause of performance or connectivity issues. To this end, we obtain the per-client statistics of all clients who do not satisfy the threshold condition. The reasons for performance issues could be manyfold: poor link quality, congestion, high losses, interference from neighboring networks or non-802.11 sources, and so on. In order to focus on the problem, we collect per-client airtimes A_i, packet loss rates L_i and overhead indices O_i. Per-client airtime A_i enables us to check for two problems. First, if the sum of A_i for all clients is less than the overall airtime A reported by the AP, this indicates external interference, and may require the administrator to perform channel-selection. Second, if the overall airtime is high, we need to check whether this is because of high congestion or poor link quality. Performance issues can also manifest in the form of lossy links. We consider a link as lossy if the loss rates exceed 5% [5]. If the loss threshold is exceeded at the second tier, it could either be because of packet collisions due to excess network congestion or a poor link.

Depending on the tier 2 metric values and triggers that are activated, the set of tier 3 metrics is selected. High airtime requires that we collect both transmission data rates (R_i) and received signal strength (S_i). Low signal strength is an indication of poor link quality. The administrator could increase the transmit power of the AP to resolve the issue. Low data rates in the presence of high signal strength indicate high congestion and the administrator could resort to admission control or load balancing to remedy the situation. On the other hand, low data rates along with low signal strength indicates poor link quality. If one client is consuming a disproportionate amount of airtime because of a poor link, the administrator could rate limit the client to resolve this condition. A lossy link in tier 2, however, necessitates only signal strength measurements at tier 3 to distinguish between losses due to congestion and losses due to poor signal strength. Finally, high overhead at tier 2 requires the collection of per-client management statistics in tier 3 to distinguish between connectivity issues. A large number of association responses or reassociation responses from the AP indicates that clients are not able to sustain a connection. This could be because of high congestion [11] wherein clients are not able to successfully associate, or they are flip-flopping between APs due to high losses. On the other hand, a large number of authentication or deauthentication messages indicates that the client authentication problems are either due to incorrect network keys or MAC address whitelisting or blacklisting.

F. Choosing Parameters

We now discuss the selection of the two parameters that influence Antler's performance: threshold values and periodicity of monitoring.

Thresholds: An important aspect of the decision tree is the choice of thresholds used in the triggers. We derive some thresholds from the network goals, e.g. minimum throughput for a connected client is obtained from the network deployment goal. Previous research guides us in the choice of some thresholds. For example, we choose an airtime threshold of 60% as this represents a moderately-congested network and allows the detection of problems before the network becomes highly-congested (airtime $> 85\%$) [12]. To detect connectivity problems, we set the overhead index at the first tier to be a very high value, indicating that clients have not been able to transmit data packets. Yang and Vaidya list the maximum achievable data rate for a given signal strength [13]; we use this as a guideline. We consider a link as lossy of the packet loss rate exceeds 5% [5]. For other metrics we plan to analyze network traces from actual networks[2] and simulations. We consider traces that contain problem scenarios as well as those that exhibit normal behavior. In this manner we are able to determine suitable threshold values for the metrics.

False Positives and False Negatives: A desirable property of Antler is to have minimal false negatives in problem identification, i.e. we do not want to miss detection of a fault. On the other hand, too many false positives (i.e. transitions to collect detailed metrics when there are no problems in reality) reduce

[2] http://www.crawdad.org

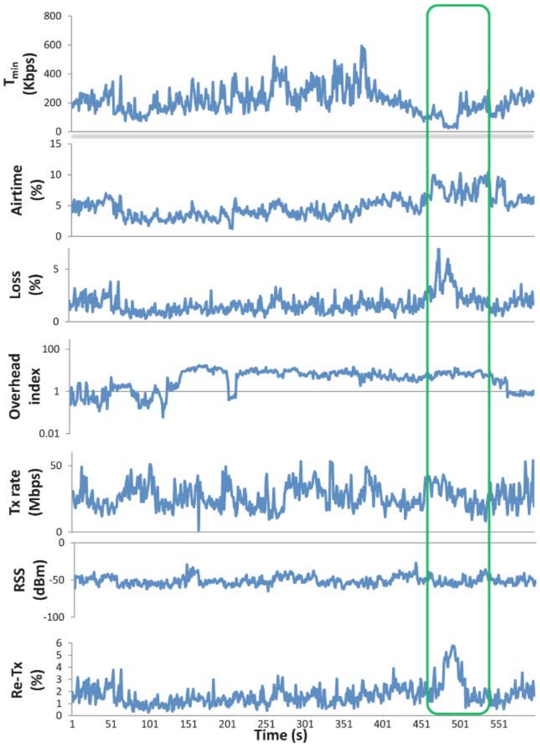

Fig. 3. Metrics taken from a 10 minute trace at the 67^{th} IETF meeting.

the effectiveness of the system in saving bandwidth. These considerations impact the choice of metric thresholds. For example, consider the airtime metric A where a high value of A might indicate a problem. A low threshold value $A_{threshold}$ leads to many false positives and frequent collection of detailed metrics. On the other hand, a high value of $A_{threshold}$ may cause the system to miss faulty conditions.

Periodicity: In contrast to a system such a Jigsaw [1] that captures every packet transmission, Antler works mainly on a statistical view of the network. Thus Antler may be unable to capture transient network conditions. Instead it focuses on more persistent problems. The response time of Antler depends on the time granularity of metric collection. A smaller period of collection makes the system more responsive to temporary conditions in the network but increases the bandwidth requirements and the workload of the central controller. A large window saves bandwidth but may cause the system to miss some network faults. In our initial design we chose to collect metrics every five seconds. We believe this value provides a balance between transient fault detection and system responsiveness.

IV. Evaluation

There are two important aspects in the evaluation of Antler. First, in Section IV-A, we show that the decision tree indeed leads us to the correct conclusions about the network problems. In order to diagnose a fault correctly, we need to make the correct choice of metric selection at each tier. Second, we show the benefits of Antler in terms of bandwidth savings in Section IV-B.

A. Design Verification

In order to verify the feasibility of multi-tiered approach to fault detection, we analyze the traces we collected from the 67^{th} IETF meeting held in November 2006 [14]. The network consisted of over 100 APs on both 802.11a and 802.11g frequencies, and was used by more than 1200 users with periods of high network utilization. Our analysis of the traces shows that the network suffered from high interference and loss rates, making it suitable to analyze the correctness of our algorithm in identifying the specified problems [14]. An example 10 minute trace is shown in Figure 3.

The figure represents the metric values for a 10 minute sample of the plenary session[3]. For the purpose of analysis, we fix the expected network throughput at 50Kbps. The low threshold is because of the high client-AP ratio, wherein we observed a single AP could attempt to serve as many as 100 simultaneous clients. At about seven minutes into the trace, we see that the throughput decreases below the minimum threshold. In a live network with Antler deployed, this would trigger the collection of second tier metrics: per-client airtime, per-client loss and per-client overhead index for the client that went below the threshold. If we examine these tier 2 metrics for the client whose throughput went below the threshold, for the same 10 minute period, we see that the client overhead does not show much variation. However, the client airtime and loss start to increase around the time throughput decreased. The increase in loss rate could be due to either the client obtaining low transmission rates or suffering from congestion, both of which lead to the increased airtime. The high loss would trigger the collection of the tier 3 metric, the client's signal strength. The RSS values plotted in the figure indicate that the AP received packets with almost no variation in signal strength. Constant signal strength coupled with high loss clearly indicates that the network was suffering from congestion. This is further verified by plotting the retransmission rate of the client (Re-Tx).

We draw two conclusions from Figure 3. The first is that the hierarchy of metrics that we described in Section III is plausible and works in a live network. This has encouraged us to further explore how we could map common wireless problems into the hierarchical decision tree. The second conclusion we draw is that we are able to confirm our hypothesis that only a subset of metrics is required to diagnose a network problem. Our algorithm provides the most useful set of metrics to be analyzed. Apart from reducing bandwidth demand, our system is the first step towards automated network management and recovery.

B. Efficiency

Having seen an illustration of how the decision tree can be used to detect network problems and discover their root cause, we now evaluate how often Antler is able to correctly diagnose network problems. For this purpose, we use the traces from the entire four hour plenary session of the IETF meeting. We compare our hierarchical approach with a naïve trace-driven

[3] In the plenary session, approximately 600 IETF attendees gathered for four hours in one room equipped with about eight 802.11g APs

approach in which the AP reports the entire set of metrics at every monitoring interval. We first use the trace-driven approach to detect all instances of potential network problems and compare this with the number of instances detected by Antler. Ideally, our system should detect all problems that are detected by the trace-driven approach, but with a much lower overall bandwidth requirement. For fairness in evaluation, our definition of a network problem remains consistent in the trace-driven approach and Antler: low throughput, high overhead or high airtime. The success of Antler depends largely on the thresholds chosen for the metrics and the periodicity of the detection engine. We use the values listed in Section III-F.

	Faults detected	Monitoring data (Mb)	False positives
Trace-driven	97	65.7	$N.A.$
Antler	84	15.4	9

TABLE I
COMPARISON OF ANTLER WITH A TRACE-DRIVEN APPROACH.

Table I shows the comparison results of Antler with a trace-driven approach. The main difference between the two approaches is in the amount of information available at any point of time to make a decision. In the trace-driven approach, we have access to all the metrics, whereas in the Antler system, we have only a subset of the metrics. Hence, in the first case we assume a fault that can be detected can also be diagnosed in one cycle of metric collection. On the other hand, Antler requires more than one collection cycle to diagnose a fault. We simulated the transitions among the different tiers based on events observed in the trace. We see that Antler was able to diagnose 85% of the faults using only about 25% of the bandwidth used in the trace-driven approach. A false positive is generated when a next tier of metric collection is triggered, but the next tier reveals no problems. We trigger data collection needlessly only for 9 instances in the entire four hour trace, which translates to about 9% error. These results are encouraging and we would like to further explore the sensitivity of the system to various thresholds to further improve the fault detection capability.

V. CONCLUSION AND FUTURE WORK

Wireless network management is challenging due to the lack of holistic network view. This work is a first step in automating wireless network monitoring and management. In this paper, we demonstrated the need for a dynamic and adaptive metric collection system. We described the design of the multi-tiered metric collection system of Antler. Our initial evaluation of the design indicated a significant reduction (about 75%) in the bandwidth requirement for network monitoring.

The promising results encourage us toward the development and deployment of Antler on a production WLAN with active usage. We hope to gain a better understanding of the performance of Antler through this deployment. In particular, we would like to explore the degree of reduction in monitoring data and the effectiveness of problem identification. Usage in a production network would also provide us with a better understanding of the metrics relevant to each network fault and the thresholds to use for these metrics.

Antler currently focuses on automated multi-tiered metric collection to assist in fault diagnosis. This can be used to bring the network administrator's attention to problems in real-time and better facilitate quick problem resolution. As part of our future work, we would like to augment Antler with automated remedial actions. In other words, the system would identify the cause of a problem and use an appropriate preconfigured solution to rectify the problem.

Another important goal and future work of Antler is to be able to perform network health monitoring in a multi-hop mesh network. Mesh networks are particularly challenging because they use wireless links for backhaul connectivity. In such networks, excessive monitoring-related traffic can consume valuable bandwidth and be detrimental to the performance of the network. Additionally, the mesh routers have to deal with an extensive set of metrics that characterize the multi-hop connectivity, in addition to the traditional WLAN-like metrics related to client access. For precisely these reasons, we believe an Antler-like hierarchical metric collection method would be particularly well-suited for mesh networks.

ACKNOWLEDGMENTS

This work is supported in part by NSF Wireless Networks award CNS-07220275 and a grant from Intel Corporation.

REFERENCES

[1] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis," in *Proc. of SIGCOMM*, Pisa, Italy, Sep. 2006.

[2] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker, "Automating Cross-Layer Diagnosis of Enterprise Wireless Networks," in *Proc. of SIGCOMM*, Kyoto, Japan, Aug. 2007.

[3] A. Jardosh, P. Suwannatat, T. Hollerer, E. Belding, and K. Almeroth, "SCUBA: Focus and Context for Real-time Mesh Network Health Diagnosis," in *To appear in Proc. of PAM*, Cleveland, OH, Apr. 2008.

[4] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker, "MOJO: A Distributed Physical Layer Anomaly Detection System for 802.11 WLANs," in *Proc. of MobiSys*, Uppsala, Sweden, Jun. 2006.

[5] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks," in *Proc. of MobiCom*, Philadelphia, PA, Sep. 2004.

[6] R. Chandra, V. Padmanabhan, and M. Zhang, "WiFiProfiler: Cooperative Diagnosis in Wireless LANs," in *Proc. of MobiSys*, Uppsala, Sweden, Jun. 2006.

[7] (2008, Feb.) Netdisco - Network Discovery and Management. [Online]. Available: http://www.netdisco.org/

[8] (2008, Feb.) AirWave Management Platform. [Online]. Available: http://www.airwave.com/

[9] (2008, Feb.) AirMagnet. [Online]. Available: http://www.airmagnet.com/

[10] (2005, Jul.) Meru Networks White Paper. [Online]. Available: http://www.merunetworks.com/form.php?typ=/technology/documents.php&file=Meru_RS_WP1-0705.pdf

[11] A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth, "IQU: Practical Queue-based User Association Management for WLANs," in *Proc. of MobiCom*, Los Angeles, CA, Sep. 2006.

[12] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding Congestion in IEEE 802.11b Wireless Networks," in *Proc. of IMC*, Berkeley, CA, Oct. 2005.

[13] X. Yang and N. Vaidya, "On the Physical Carrier Sense in Wireless Ad Hoc Networks," in *Proc. of INFOCOM*, Miami, FL, Mar 2005.

[14] R. Raghavendra, E. M. Belding, K. Papagiannaki, and K. C. Almeroth, "Understanding Handoffs in Large IEEE 802.11 Wireless Networks," in *Proc. of IMC*, San Diego, CA, Oct. 2007.

978-1-4244-2219-7/08/$25.00 (c)2008 IEEE