MeshMon: A Multi-tiered Framework for Wireless Mesh Network Monitoring

Ramya Raghavendra, Prashanth 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

Monitoring and troubleshooting a large wireless mesh network presents several challenges. Diagnosis of problems related to wireless access in these networks requires a comprehensive set of metrics and network monitoring data. Collection and offloading of a large amount of data is infeasible in a bandwidth constrained mesh network. Additionally, the processing required to analyze data from the entire network restricts the scalability of the system and impacts the ability to perform real-time fault diagnosis. To this end, we propose MeshMon, a network monitoring framework that includes a multi-tiered method of data collection. MeshMon dynamically controls the granularity of data collection based on observed events in the network, thereby achieving significant bandwidth savings and enabling real-time automated management. Our evaluation of MeshMon on a real testbed shows that we can diagnose a majority (87%) of network faults with a 66% savings in bandwidth required for network monitoring.

Categories and Subject Descriptors: C.2.3 [Computer - Communication Networks]: Network Operations: Network monitoring; Network management

General Terms: Experimentation, Management, Measurement, Performance.

Keywords: Wireless network management, mesh network, heirarchical, wireless troubleshooting.

1. INTRODUCTION

Large scale IEEE 802.11 mesh networks promise to be a significant method of providing Internet connectivity in several cities and towns. In addition to these metro-scale deployments, wireless mesh networks (WMN) have been proposed to provide connectivity in rural environments, especially in developing countries around the world. The presence of numerous wireless devices,including mesh routers and client devices, in a single administrative domain increases the complexity of managing these large scale mesh networks.

We believe the network administrator's ability to manage and troubleshoot these networks in real-time is a critical factor that contributes to the success of WMNs. These administrative tasks, however, present several new challenges compared to traditional wireline networks. In particular, the design of a network monitoring system is non-trivial because of the multi-hop architecture of these mesh networks and the inherent wireless-related properties of 802.11-based devices. Several factors contribute to the challenges of effective monitoring and management of mesh networks. For instance, the performance of the devices in these 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. Unlike in WLANs, the backhaul links used for communication between mesh routers and the Internet Gateway consist of relatively low bandwidth multi-hop wireless links. Therefore, control traffic required for remote monitoring and administration of these mesh routers must be minimal, so as not to consume a significant portion of the available bandwidth. Finally, unlike wired networks, the physical location of the mesh routers provides a strong spatial aspect to all data used in management and troubleshooting of mesh networks. Therefore, data from different routers that share spectrum in a geographical region may need to be analyzed in correlation with each other.

Due to the inherent uncertainty in the wireless medium, network administrators require a comprehensive set of data and metrics to deal with them. The data include metrics from the 802.11 MAC layer and the PHY layer, in addition to those from higher layers of the stack including routing, network topology, etc. Monitoring of Although traditional infrastructure WLANs present similar monitoring challenges and requirements, network monitoring solutions developed for WLANs cannot be directly applied to WMNs. Most monitoring solutions for commercial WLANs only use a small fixed subset of the large set of available metrics to minimize the data collection and processing overhead. This approach may fail to capture data needed to diagnose a detected problem. 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 remote analysis of all data packets is infeasible in a mesh network as the bandwidth requirements are prohibitive. Further, monitoring systems that use a large set of metrics (or detailed packet traces) require resource intensive computation and thus may be unsuitable for real-time identification and remediation of 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 visual rendering of the data, is the computational bottleneck [3].

For the above reasons, there is a need for a methodology of monitoring and metric collection in WMNs that is bandwidth-efficient, scalable with respect to the number of devices in the network, and able to provide a comprehensive set of metrics that can be used to identify all problems in the network. Such a solution 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.

MobiHoc S3'09, May 18, 2009, New Orleans, Louisiana, USA.
ACM 978-1-60558-521-5/09/05.

In this paper, we present MeshMon, a network monitoring framework that enables real-time identification and troubleshooting of problems in WMNs. A key observation that guides the design of MeshMon is that comprehensive metric collection is required only when there are problems in the network. A small subset of these metrics, called baseline 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. When the baseline metrics indicate the possible presence of a problem, the system transitions to collect a more detailed set of metrics. The goal of this methodology of metric collection is to reduce the volume of data that needs to be collected and processed without sacrificing the ability to diagnose problems in the network.

In this work we develop the idea dynamic and scalable hierarchical metric collectionin the context of mesh networks. Mesh networks offer additional complexity as compared to WLANs because a monitoring system should address problems that affect mesh routers as well as those that affect client devices. Therefore, MeshMon incorporates metrics associated with mesh routing and connectivity into the hierarchical metric collection, in addition to metrics associated with client devices. Our design ensures that even in situations where a problem scenario is reflected in both sets of metrics (mesh related and client access related), MeshMon can successfully isolate the root cause of the problem.

Our contributions in this work are as follows:

- We present a classification of WMN metrics in a hierarchical structure to assist in automated fault diagnosis.
- We present the detailed design, implementation and evaluation of the entire MeshMon system.
- We have implemented a prototype of MeshMon on the UCSB MeshNet testbed.[1] The prototype system is capable of identification and diagnosis of a variety of common problems that occur in WMNs. Our evaluation of MeshMon indicates that we diagnose 86.7% of the problems with a 66% reduction in the bandwidth required for monitoring data.

2. DESIGN OF MeshMon

Our solution is designed for a multi-hop IEEE 802.11-based mesh network. Some aspects of the baseline design were first presented in our previous work that was designed for WLANs [4]. The network architecture we assume is that each mesh router is equipped with two radios - one used for the backhaul connectivity to the Internet, and the second radio as an AP that services client 802.11 devices.

The basic idea in the design of MeshMon 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 metrics in the area where the problem is detected can facilitate fine-tuned problem diagnosis.

In the design of MeshMon, 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.

[1] http://moment.cs.ucsb.edu/meshnet

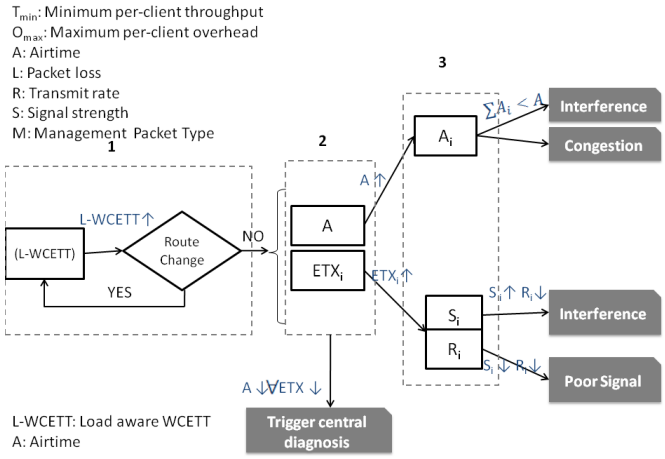

Figure 1: Multi-tiered metric collection decision tree for the mesh backhaul implemented in the analysis engine. The numbers at the top indicate the tier of metric collection. White boxes represent the metrics collected at each tier. Arrows indicate the triggers used to transition between tiers. Black boxes indicate the fault diagnosis.

To select a baseline set of metrics, we consider the typical performance goals of a mesh network [5]. Broadly, there are two goals that a wireless mesh network tries to achieve: 1) provide connectivity to clients within the network's coverage area, and 2) ensure high quality routes to the gateway. We note that ubiquitous coverage is a goal during the deployment phase of a network; we are concerned with detecting performance issues during post-deployment operation. Therefore the ultimate objective is to ensure clients are able to connect to the network and obtain good performance from the mesh network.

These objectives lead us to three baseline metrics: maximum client overhead index (O_{max}), load-aware WCETT (L-WCETT) and minimum client throughput (T_{min}). Overhead index is defined as the ratio of control and management traffic to data traffic [6]. When a client has connectivity problems, O_{max} will be high. The second metric, L-WCETT, provides a measure of the mesh performance. WCETT [7] is a metric used for making routing decisions in a mesh network and was originally designed to be load independent, since a routing algorithm should be resilient to route flapping. However, for the purpose of detecting network faults, we would like to be able to account for throughput degradation due to network load. We do this by adding to WCETT the queuing and contention delays along the path. We call this metric load-aware WCETT (L-WCETT). L-WCETT measured at a mesh router is closely related to the mesh path throughput achieved between the mesh router and its gateway. A high value of L-WCETT indicates reduced path throughput for a mesh router. The third baseline metric, T_{min}, tracks the performance of connected clients. When a client obtains low throughput, T_{min} will be low.

Figure 1 presents the visual representation of the hierarchy tree for the mesh backhaul. Similar hierarchy exists for the client layer of the mesh and is presented in detail in the full version of the paper.

Trace	CBR	Replay
Faults Injected	30	30
Faults Detected	27	25
False Positives	8	10
Overhead Reduction	68%	64%

Table 1: Fault diagnosis performance of MeshMon.

In the normal state of network operation, MeshMon operates in Tier 1 and the Current Set is comprised of the baseline metrics.

When a baseline metric for the mesh layer crosses its threshold, only the mesh layer decision tree is activated. However, when a problem is detected in the client layer, both the client access tree and the mesh access tree are activated, since the fault could lie in either tier. If the problem lies in the client layer, all information required for diagnosing is present locally and hence fault diagnosis occurs on the mesh node. However, when the fault lies in the mesh layer, a mesh node attempts to locally diagnose a problem. If unsuccessful, it contacts the gateway, which in turn will turn on the diagnosis on the mesh nodes on the node's upstream path.

As an example, consider the scenario where a mesh node's throughput has dropped below the threshold because a mesh router further up the route to the gateway is congested. Tier 2 of metric collection is triggered to collect the node's local airtime and ETX metrics. Both these metrics would not indicate the problem that lies upstream. At this point, the mesh router triggers the gateway to initiate centralized diagnosis, and the gateway triggers the collection on each mesh router along the node's upstream path. Congestion will be detected by the gateway since $\sum_{i=0}^{n} LETT_i < LETT_t$ where $LETT_i$ is the per-link load aware ETT (L-ETT) value along the path and L-ETT_t is the threshold computed.

3. EVALUATION

The system is evaluated by injecting faults into the network and comparing the number of faults detected with the number injected.

A prototype of the MeshMon system has been implemented on the UCSB Meshnet. The implementation involves simple extensions to the madwifi driver as well as software at the user level.

Evaluations are conducted with two types of traffic: a) constant rate flows which we call the CBR traffic, and b) traces from a large WLAN, which we call the Replay traffic. In each of the scenarios, eight laptops act as clients connected to the UCSB mesh network. In the first scenario, each laptop sends CBR traffic at a constant rate of 1Mbps to the gateway. In the second scenario, we use the WLAN traces collected from the IETF 67 wireless network to extract link layer data traffic patterns and use this information to replay the traffic on the mesh testbed [8]. [2]

Our general evaluation methodology is as follows. We inject a set of faults into the system. The nodes run MeshMon and attempt to diagnose the faults through increased metric collection and send alerts to the central controller when the fault is detected. We quantify the diagnosis accuracy by comparing the inferred fault and its source with the original fault we injected. We inject faults in both the client access layer and the mesh layer.

Fault diagnosis accuracy and overhead reduction: The complete set of results from the experiments is presented in Table 1. Of the total 60 faults injected in the two scenarios, MeshMon successfully detected 52. The average reduction in overhead for the two scenarios was 66%. In other words, MeshMon was able to detect a high percentage (86.6%) of faults using only one-third of the monitoring bandwidth as compared to the simple approach of using all the available metrics. For our simple testbed setup with 15 nodes and a maximum of one client per mesh node, the simple monitoring approach collected about 400MB of monitoring data for a four hour period, while MeshMon required about 134MB. This is an encouraging result that indicates that MeshMon can scale better and can support larger mesh networks.

[2]A detailed description of both scenarios is in a full version of this extended abstract.

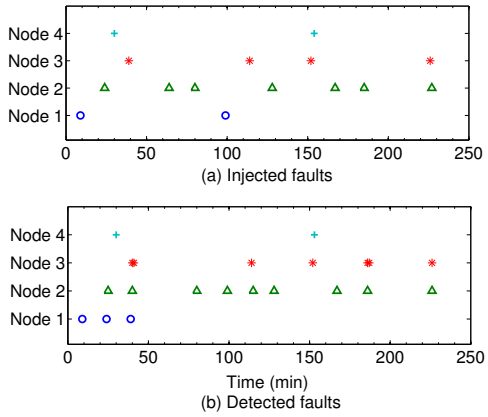

Figure 2: Time series of faults injected and detected at the mesh layer in a representative experiment trial.

The results in Table 1 indicate a high number of false positives and hence we further investigate this behavior. We observe that for some injected faults, the central controller receives alerts from multiple mesh routers. MeshMon currently does not have the capability of correlating alerts posted by multiple mesh routers. Such a capability would enable MeshMon to distinguish a fault that simultaneously impacts the performance of multiple mesh routers and reduce the misleading false positive rate.

4. 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 *Proc. of PAM*, Cleveland, OH, Apr. 2008.

[4] R. Raghavendra, P. A. K. Acharya, E. M. Belding, and K. C. Almeroth, "Antler: A Multi-Tiered Approach to Automated Wireless Network Management," in *Proc. 1st IEEE Workshop on Automated Network Management*, Phoenix, AZ, Apr. 2008.

[5] J. Robinson and E. Knightly, "A Performance Study of Deployment Factors in Wireless Mesh Networks," in *Proc. of INFOCOM*, Anchorage, AK, May 2007.

[6] 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.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-radio, Multi-hop Wireless Mesh Networks," in *Proc. of MobiCom*, Philadelphia, PA, Sep. 2004.

[8] 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.