Pervasive and Mobile Computing 6 (2010) 93–111

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Fast track article

A study of overheads and accuracy for efficient monitoring of wireless mesh networks

Dhruv Gupta *, Daniel Wu, Prasant Mohapatra, Chen-Nee Chuah

Graduate Group in Computer Science, University of California Davis, Davis, CA 95616, United States

ARTICLE INFO

Article history:
Received 13 February 2009
Received in revised form 2 July 2009
Accepted 13 July 2009
Available online 17 July 2009

Keywords:
Wireless mesh networks
Monitoring
Measurement
Accuracy
Overheads

ABSTRACT

IEEE 802.11-based wireless mesh networks are being increasingly deployed in enterprize and municipal settings. A lot of work has been done on developing measurement-based schemes for resource provisioning and fault management in these networks. *The above goals require an efficient monitoring infrastructure to be deployed, which can provide the maximum amount of information regarding the network status, while utilizing the least possible amount of network resources.* However, network monitoring involves overheads, which can adversely impact performance from the perspective of the end user. The impact of monitoring overheads on data traffic has been overlooked in most of the previous works. It remains unclear as to how parameters such as number of monitoring agents, or frequency of reporting monitoring data, among others, impact the performance of a wireless network. In this work, we first evaluate the impact of monitoring overheads on data traffic, and show that even small amounts of overhead can cause a large degradation in the network performance. We then explore several different techniques for reducing monitoring overheads, while maintaining the objective (resource provisioning, fault management, and others) that needs to be achieved. Via extensive simulations and experiments, we validate the efficiency of our proposed approaches in reducing overheads, their impact on the quality of data collected from the network, and the impact they have on the performance of the applications using the collected data. Based on results, we conclude that it is feasible to make the current monitoring techniques more efficient by reducing the communication overheads involved while still achieving the desired application-layer objectives.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

IEEE 802.11-based wireless mesh networks (WMNs) have witnessed a tremendous growth over the last few years [1–3]. A lot of work has been done in terms of understanding the behavior of these networks and analyzing the impact of various factors, such as interference and number of hops, on their performance [4–6]. A lot of focus has also been on providing Quality-of-Service (QoS) in WMNs and several schemes have been proposed for admission control and QoS-based routing [7–10]. There have been some recent studies on developing measurement-based schemes for resource management and fault management in WMNs [7,11]. These schemes usually involve measuring certain parameters from the network (such as packet loss rate or signal quality), and utilizing this data for the purpose of QoS provisioning (for example routing algorithms using loss rate as metric) or for fault management (for example avoiding links with low signal quality). WMNs have also found wide applications in enterprize and municipal networks. Such networks need to be monitored constantly for performance degradation and other anomalies. Network operators would like to have an efficient monitoring framework

* Corresponding author.
E-mail addresses: dhgupta@ucdavis.edu (D. Gupta), danwu@ucdavis.edu (D. Wu), pmohapatra@ucdavis.edu (P. Mohapatra), chuah@ucdavis.edu (C.-N. Chuah).

1574-1192/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2009.07.011

that can provide them with up-to-date network statistics. This is not a trivial task and the challenges involved in network management and diagnosis have been addressed in the past [12–14].

Both the above goals (QoS provisioning and network diagnosis) require an efficient monitoring framework that can provide accurate statistics about the network in a timely manner. An online network management system would require transmitting measurement data from various locations to a central server, or the exchange of data among various mesh nodes. *However, in most cases, the same links are used for carrying both the user traffic and the monitoring data.* In the case of 802.11-based wireless networks, an out-of-band channel may not always be available for transferring measurement data from the mesh nodes to a central server. Deploying dedicated monitoring sniffers, with each node having a connection to the wired backbone, is cost-prohibitive and may not always be feasible. As a result, the transmission of measurement information will contend with the data traffic and reduce the channel time available to the end users. As a result, in a multi-hop wireless network, even small amounts of monitoring traffic can cause a large impact on the existing data traffic in the network, resulting in performance degradation for the end users. Thus, reducing the amount of monitoring overhead in a wireless network is an important goal.

The traditional approach for deploying monitoring infrastructures has been to use a network-wide periodic measurement framework. An ideal case would be where each node in the wireless mesh network also acts as a monitoring agent, periodically reporting measurement data to the central server. Such an approach gives us the advantage of having an accurate and up-to-date image of the network status. However, a network-wide monitoring infrastructure reporting periodic data may introduce large amounts of overhead in the network. In order to achieve our objective of reducing monitoring overheads, we propose the concept of application-based monitoring. Our conjecture is that different monitoring frameworks can be customized based on the application that uses the measurement data collected from the network. As part of this study, we consider two different application scenarios and show how different techniques lend themselves to each scenario:

- *QoS provisioning:* A municipal ISP may have different service classes for its users, where each service class offers certain guarantees in terms of network performance (such as bandwidth and delay). In order to make sure that each user gets the specified performance, it is necessary for the ISP to continuously monitor the network. For example, the ISP may infer end-to-end delay of a particular path by collecting delay information from the wireless routers along that path, and then utilize this information to provide delay guarantees to end users. In such a scenario, each wireless router needs to periodically report a certain set of parameters to a central server, where this information will be analyzed. For such applications, we propose two different monitoring solutions:
  - *Monitor Selection approach:* In this approach, we propose to reduce the overheads by reducing the number of wireless routers used as monitoring agents in the wireless mesh network. We use a vertex-cover algorithm to locate the mesh nodes that should also serve as monitoring agents. This helps us to reduce the number of monitors in the wireless network, while still being able to maintain complete link coverage.
  - *Reporting Interval approach:* Our second approach involves decreasing the frequency with which we report data to the central server. By sending out monitoring packets at longer intervals, we reduce the contention in the network.
- *Network diagnosis & fault management:* An enterprize network administrator may be interested in maintaining the performance of the entire network above a certain level, as opposed to providing service guarantees to individual users. An efficient monitoring framework is required to report any events that might indicate a degradation in the network performance. For example, the administrator may be interested in monitoring the signal quality on different links, in order to decide whether a link should be used or not. In such a scenario, the administrator may not be interested in periodic reports, but would instead like to get a measurement value if and only if the signal quality of a link falls below a threshold value. For such applications, we propose the following approach:
  - *Threshold-based monitoring:* In this approach, the nodes serving as monitoring agents will transmit measurement data only when a certain pre-defined event occurs, as opposed to a periodic framework that reports data on a continuous basis. An example of an event would be a desired network parameter crossing a pre-defined threshold.

We propose these techniques not as a replacement for existing network monitoring solutions, but to complement these schemes by making them more efficient in terms of overhead. *However, a simple reduction in network overheads is not our final goal.* It is also crucial for us to investigate as to what impact does that have on the **quality of information** that is being collected from the network. This is important because if our accuracy of estimating network statistics goes down, then this will impact the performance of the application that is using the collected data. Hence, for each of the above techniques, we also evaluate the trade-off involved in terms of reduction in overheads and the accuracy and quality of measurement data. This analysis is of utmost importance, as a reduction in overheads will not be desirable if it causes a large degradation in the quality of measurement data, and the performance of the application using that data. In our previous work [15], we performed some basic simulations to test our proposed approaches. In this work, we perform more extensive simulations, and also validate our schemes via extensive experiments on our laboratory test-bed. *Our main contributions can be summarized as follows:*

- **We investigate the effects of monitoring overheads on the forwarding of user data traffic. We show that even small amounts of monitoring traffic can result in increased delays and packet loss for the end user.**
- **We propose and evaluate three different approaches: (a) monitor selection approach, (b) reporting interval approach, and (c) threshold-based monitoring, for reducing overheads in WMNs. Instead of using one common**

solution, we propose the concept of application-based monitoring, which involves using different monitoring solutions for different applications.

- **Our most important contribution involves evaluating the trade-offs between estimation accuracy and reduction in overheads. We evaluated the accuracy of estimating certain network parameters and its impact on the performance of the application using the data.**

The rest of the paper is organized as follows. Section 2 outlines some of the previous work in this area, and the motivation behind our work. Section 3 explains the proposed methodology. Section 4 gives the performance evaluation of the proposed schemes, along with the simulation results. Section 5 outlines the results obtained from our experimental study. Section 6 concludes the paper.

2. Related work and motivation

2.1. Related work in wired networks

The problem of efficient monitoring in a wired network (such as the Internet backbone) has been studied in the past. Several works have studied how to use different polling mechanisms for lowering overheads [16,17]. Other works have looked at improving the performance of reactive monitoring in wired networks. In [18] and [19], the authors look at how to combine global polling with local event-based reporting for reducing monitoring overheads. However, these works did not consider the impact of using these mechanisms on the functionality to be achieved. Other works such as [32] and [34] have considered the impact of reducing the frequency of routing updates on routing overheads, as well as on end user performance. However, such works focus only on the impact of delaying or reducing routing updates, while our goal is to reduce the amount of monitoring data and evaluate its impact on the performance of a broader class of applications.

Several recent works have studied this problem from the aspect of jointly reducing the number of monitors and controlling the sampling rate at the monitors, in order to bring down the monitoring cost while maximizing the monitoring coverage in terms of the number of flows monitored. In [20], the authors consider the problem of minimizing cost (sum of deployment and monitoring cost) and maximizing coverage (in terms of monitoring reward) under various budget constraints. In [22], the authors look at the problem of placing a small set of active beacons in the Internet topology. They show that the problem is NP hard for a BGP-like routing topology and present the upper and lower bounds for the number of beacons needed for a given network. In [23], the authors present active monitor placement as a combinatorial problem and present a mixed integer programming solution. They propose algorithms to both minimize the number of monitoring beacons and the sampling rate. In [24], the authors consider the problem of placing monitors and setting the sampling rate. Like the previous works, they show the problem to be NP hard and present approximation algorithms to solve the problems. In one of the most recent works [21], the authors propose that the monitor placement should be a dynamic process, based on the variations in the network traffic. They propose an approach where a monitor is assumed to be present at every network node. The problem is to decide which monitors to activate and what sampling rate to set at each monitor, in order to achieve a measurement task with high accuracy and low resource consumption.

Unlike the above works, our focus is on multi-hop wireless networks. The primary difference that arises between these two scenarios is the definition of the "cost" involved in network monitoring. In wired networks, the focus has been to minimize the deployment and maintenance cost of the monitors (especially given the large size of the networks), while the communication overheads usually do not present a problem. Second, since the number of links and flows to be monitored is huge, choosing a sampling rate along with the placement of monitors becomes crucial. Hence, all the above works approach the monitor placement problem in wired networks as a joint problem of minimizing cost and maximizing coverage, and have presented heuristic solutions. On the other hand, in wireless networks, communication overheads pose a greater problem due to interference and limited available bandwidth. By controlling the measurement overheads, we can increase the channel time available to the end users. Hence, instead of looking at deployment and maintenance costs, we define the "cost" for monitoring wireless networks to be in terms of the accuracy of measurement. We evaluate the proposed schemes by studying their impact on the accuracy with which we can monitor data, and the impact the schemes have on the application using the measurement data. Moreover, due to the smaller size of wireless networks, and lower link speeds, it is usually not required to set a sampling rate on a wireless monitor. Hence, the need for different efficient monitoring techniques for wireless networks.

2.2. Related work in wireless networks

Wireless networks present a widely different scenario than wired networks. A wireless mesh network does not compare to the Internet in terms of network size, link speed, and number of flows. Hence, sampling rates do not need to be determined for each monitoring node. Also, wireless networks have highly dynamic characteristics (interference and varying link quality), which should be taken into account during the deployment of the monitoring infrastructure. Several previous works have proposed both active probe-based and passive monitoring techniques for wireless mesh networks. However, these works do not focus on the impact that monitoring overheads have on the transmission of data traffic. For example, routing schemes based on metrics such as ETX [11] and ETT [4] rely on periodic broadcasts to estimate the link quality.

However, the impact of overheads on data traffic has not been quantified in these works. In [25], authors have evaluated the impact of overheads associated with ETT-based routing. It was shown that the active probes used by ETT-based routing protocol contend with the data flows for channel access and result in reduced throughput for the end users. Passive monitoring techniques such as those proposed in [13] and [14] provide a better alternative to active monitoring. However, even with such approaches, the transmission of measurement data from the monitoring nodes to the central server will add overheads, which is not accounted for in these works. In [26], the authors perform some basic analyses to show how low-rate management and control traffic can severely degrade the end user's throughput, and propose a metric to capture this reduction in throughput. Our work extends these studies in two ways. We investigate the impact of various factors such as reporting interval and size of monitoring packet on measurement overheads. We further perform a careful analysis of how reducing overheads impacts the quality of measurement data being collected from the network and the performance of the application using that data.

Some other works such as [33] (and references thereof) have focussed on mobile Adhoc networks, and have evaluated the impact of collecting coarse-grained mobility information on QoS provisioning and overheads. Our focus is on collection of network-performance-related metrics in static wireless mesh networks, and their impact on the application using the collected data.

2.3. Motivation

As explained in the previous sections, the issue of efficient monitoring in multi-hop wireless networks is an important problem. Our goal is to evaluate different techniques that can help reduce the volume of monitoring data in the network, while achieving the desired performance and functionality. Lower monitoring traffic will translate to lower contention and interference in a wireless network, thereby providing better end-to-end performance to the clients. However, we not only evaluate techniques for reducing the monitoring traffic in the network, but we also evaluate their impact on the accuracy of measurement information and performance of various applications. *Our work is complementary to the existing network monitoring approaches for wireless mesh networks. We do not propose a new monitoring scheme itself, but instead focus on how these various monitoring techniques can be made more efficient, by reducing the overheads involved in the collection and transmission of measurement data, and what impact would this have on the accuracy of the data.*

3. Proposed methodology

Our conjecture is that different application scenarios will need different forms of monitoring. Based on the objective for which the monitoring data is being used, different techniques for reducing the volume of measurement traffic can be used. As part of this study, we consider two different application scenarios, namely QoS provisioning and Network Diagnosis (outlined in Section 1). These are only two of the several different application scenarios (such as QoS routing, load balancing, admission control, fault detection, anomaly detection and so on) that require a monitoring infrastructure. In each of the above scenarios, the volume of measurement data generated by the monitoring framework will directly impact the performance of the end users.

3.1. Monitor selection approach

For the first application of QoS provisioning, we consider QoS-based routing. We consider a delay-based routing algorithm, where the objective is to find a path with minimum delay for each client. In order to achieve this, we need to monitor each link in the network, and report the associated delays to the central management server. The central server would then use this information to estimate the end-to-end delays for various network paths. By utilizing this information, the central server can assign a path with the least delay to an incoming client.

In an ideal case, every node in the wireless mesh network would also be used as a monitoring agent. Using such a framework would enable us to collect continuous measurement data from every link in the wireless network. However, such a framework would also introduce large volumes of monitoring traffic in the network, thereby adversely affecting the transmission of data traffic. *In order to reduce the monitoring overheads, we propose to limit the number of wireless routers used for monitoring purposes, while still achieving the goal of delay-based routing.*

We decided to evaluate the performance of vertex-cover algorithm for this purpose. We use this algorithm to locate the network sites to be used for monitoring purposes. A mesh network can be modeled as a graph $G = (V, E)$, where V is the set of nodes, representing the mesh access points, and E is the set of edges, representing the links between the mesh access points. We want to choose a set of k nodes, from N nodes in the network, to be used for monitoring. The above problem is similar to the vertex-cover problem in graph theory. For our problem, if we can find a vertex cover for our network, then we have a set of nodes which we can use as monitoring agents. This would ensure that we cover all the links in the network for the purpose of monitoring, while using the minimum possible number of nodes.

A simple approximation to the vertex-covering algorithm consists of picking a random edge from the graph and adding the vertices of the edge to the vertex cover. It then removes all the edges incident on these two vertices, as they have been covered, and then repeats the above process. The running time of this algorithm is $O(V + E)$. This algorithm is a polynomial-time 2-approximation algorithm [27]. *However, we should not select any random network site to be used as a monitoring agent.*

The selection process should take into account some network characteristics. We include the effects of network topology in the monitor selection decision. To do this, we use the vertex cover approximation algorithm that chooses vertices in decreasing order of their degrees. The rationale behind this approach is that the vertex with the maximum degree would reflect the node that has the maximum number of links with other nodes in the mesh network. Thus, by choosing the nodes with higher degrees, we will be monitoring a larger number of links in the network.

3.2. Reporting interval approach

The second approach that we propose for the QoS provisioning application is the Reporting Interval approach. Once again, our goal is to implement a delay-based routing scheme that involves monitoring the per-hop delay at each mesh node.

An important parameter in any monitoring framework is the frequency at which we monitor data. Reporting data at a high frequency (such as per-second basis or lower) enables us to maintain a more accurate image of the network. However, this approach suffers from high overheads, since a large number of monitoring packets are being sent out by each mesh node. We consider a basic monitoring infrastructure that transmits one monitoring packet every second from each mesh node. In order to reduce the overheads, we can use a framework where nodes maintain an average of various parameters and report data at longer intervals. For example, instead of transmitting one monitoring packet every second, we can transmit one packet every ten seconds. This approach will result in fewer monitoring overheads being generated in the network. However, the reporting interval should be selected appropriately, so as to not impact the desired functionality. We investigate the performance of our delay-based routing protocol using different reporting intervals and evaluate the trade-offs in terms of overheads and accuracy of monitored data.

3.3. Threshold-based monitoring

For the second application of detecting performance degradations in the network, we propose to use a threshold-based monitoring framework in order to reduce the volume of monitoring traffic. Our objective here is to report data to the central server only when a certain event occurs. Specifically, in our case, we consider the traffic load on a node as an indication of congestion. If this load value crosses a certain threshold, then it might indicate an onset of congestion, and the administrator may want to route packets around that node. We wish to capture this particular event in the network. Ideally, we could use the periodic reporting mechanism, where every wireless router continuously sends monitoring data to the central server. For our case, such a framework would transmit data irrespective of the traffic load on the mesh node. However, this would generate large amounts of overhead and add to the contention in the network. We thus evaluate the performance of the threshold-based reporting mechanism to achieve our objective of reducing monitoring overheads while maintaining the desired functionality of identifying network performance anomalies.

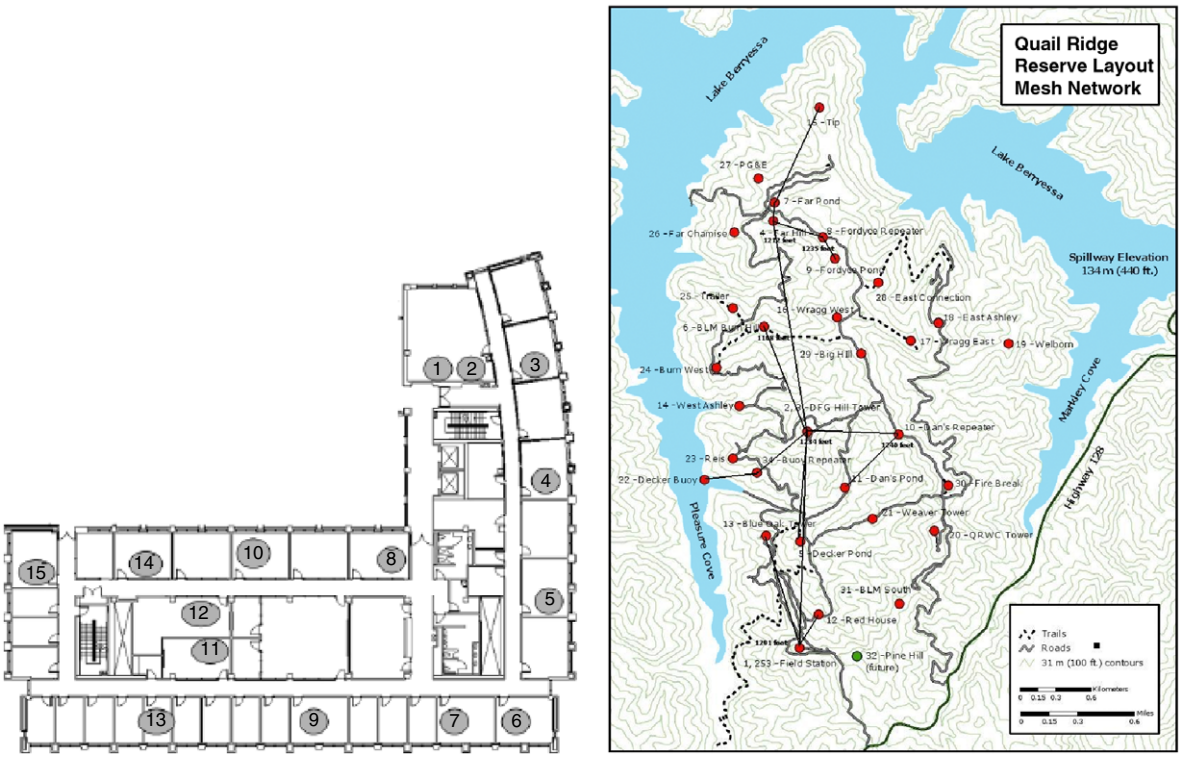
It should be noted that the threshold-based monitoring framework may not be suitable for QoS provisioning. In order to provide performance guarantees to end users, in terms of throughput or delay, we need continuous and periodic measurements from the network. A threshold-based scheme will only report data when a certain event occurs and may not be able to provide the fine-grained measurements required for QoS provisioning. We could modify the threshold in order to report data more frequently. However, this might lead to large overheads in the network. Second, consider the case where a transient performance degradation occurs in the network. A periodic monitoring framework will accumulate data and report an average value at each reporting interval. On the other hand, a threshold-based framework will transmit the instantaneous value at every occurrence of the event. As a result, the latter case might cause the QoS application to over-react to transient events, resulting in other problems such as route-flapping. *Hence, a threshold-based approach may not be suitable for QoS provisioning, and we need different types of monitoring schemes for different applications.*

4. Performance evaluation

In this section, we describe the details of our simulations, followed by an analysis of the proposed approaches.

4.1. Simulation setup

In order to evaluate our proposed scheme, we have used the QualNet simulator. We use three topologies for comparing the performance of various schemes. The first one is a twenty-five node uniform grid topology generated in QualNet (hereafter referred to as the *GRID* topology). The distance between the nodes was set to be 100 m, while the communication range was set to 200 m. We used the two-ray path loss model for this topology. The center node was selected as the gateway node. The second topology (Fig. 1(a)) is a fifteen node network, derived from the indoor testbed used in [28] (hereafter referred to as the *INDOOR* topology). It consists of 802.11 b/g nodes spread across two floors in a department building. We used the indoor terrain model library in QualNet to simulate the indoor building environment. The width of the simulation environment was set to be the same as the dimensions of the actual building which houses the testbed (about 187 feet). Node 8 in the figure served as the gateway node for this topology. The third topology is also a fifteen node mesh network, based on our outdoor mesh network testbed (hereafter referred to as the *OUTDOOR* topology). This testbed [2] consists of



(a) INDOOR topology (image from [28]). (b) OUTDOOR topology.

Fig. 1. Simulation topologies (the GRID topology is not shown).

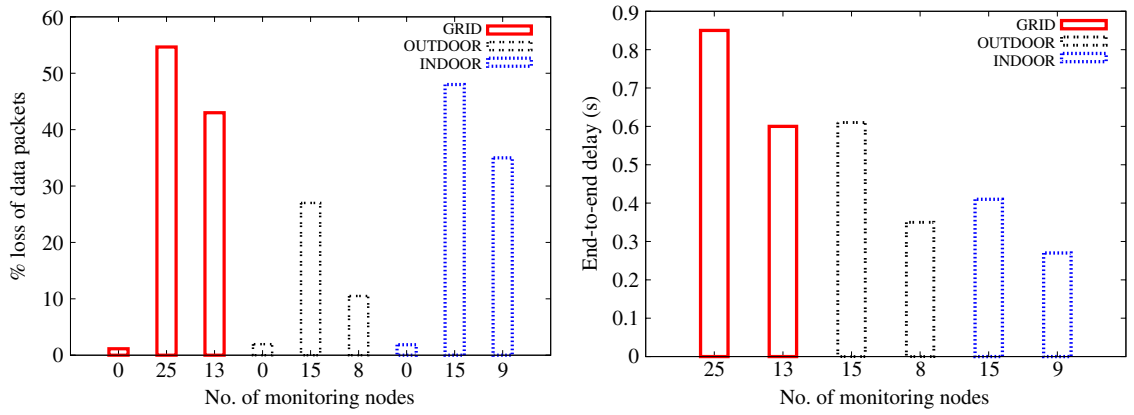
802.11 b/g nodes spread over two thousand acres in a wild-life reserve. The complete testbed is shown in Fig. 1(b), while the actual nodes used in the simulation topology are shown as connected with solid lines in the figure (the network was expanded after the completion of this work). Node 1 (field station in Fig. 1(b)) served as the gateway node for this topology. The outdoor terrain library in QualNet was used to simulate this testbed as accurately as possible. The distances between the nodes, their transmission power, and other network parameters, were set in accordance with the measurements obtained from the actual testbed.

For *GRID* and *OUTDOOR* topologies, we had six traffic sources placed uniformly across the mesh network, generating FTP data. For *INDOOR* topology, we had three traffic sources placed uniformly across the topology. In all three cases, the data traffic was destined for the gateway node, and the amount of data generated was the same. The data packet size was set to 1500 bytes. The physical transmission rate of every node was fixed at 2 Mbps, and all the nodes were operating on the same channel. RTS/CTS was disabled (as is the case with the actual testbeds). Ad-hoc On-demand Distance Vector (AODV) routing protocol [29] was used as the routing protocol of choice. We did not experiment with different configurations of traffic sources or the gateway location. Finding the optimal choice of gateway node for our schemes was beyond the scope of this work. Even if the placement of the gateway node is optimized, the monitoring nodes still need to forward measurement data to the server. Therefore, even though the absolute values of the measurement overheads may be different, the overhead cost will still exist and we expect to see similar trade-offs between estimation accuracy and overheads.

We set the default reporting frequency for the monitoring agents at one packet per second. This means that each monitoring node will collect measurement data for every packet it hears each second, at the end of which it will send a monitoring packet with the collected data. We assume that all the data has to be sent to a central server, which is co-located with the gateway node. In a practical network, this server could be the network operation center where the administrator can collect and analyze all the data on-the-fly or it could be a server for storing measurement data, which can be used later for off-line analysis. The monitoring data is sent using UDP at the transport layer. The following sections describe the various results.

4.2. Monitor-selection approach

In this section, we present the results for the monitor selection approach. As described earlier, we use a degree-based approximation algorithm for finding vertex-covers to locate the wireless routers to be used as monitoring agents. The max-degree VC algorithm returned a thirteen node vertex cover for *GRID* topology, eight node vertex-cover for *OUTDOOR* topology and a nine node vertex-cover for the *INDOOR* topology.



(a) Packet loss. (b) End-to-end delay.

Fig. 2. Impact of number of monitoring agents on user performance.

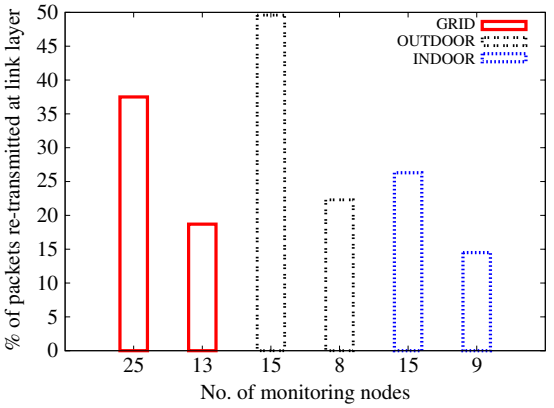


Fig. 3. Comparison of link layer re-transmissions for monitor-selection approach.

We first evaluated the impact of monitoring overheads on the data traffic, in order to justify the need for reducing monitoring overheads in WMNs. We specifically look at how the periodic reporting of monitoring information to a central server impacts the flow of user data, in terms of packet loss and end-to-end packet delay. Fig. 2(a) shows that when no monitoring agents are used in the network, the associated losses are minimal. This means that the network is not saturated to begin with. Fig. 2(a) also shows the percentage loss of data packets, when all the nodes in the topology are used as monitoring agents (25, 15 and 15 for the *GRID*, *OUTDOOR* and *INDOOR* topology), and when only the nodes in the vertex-cover set are used as monitoring agents (13, 8 and 9 for the *GRID*, *OUTDOOR* and *INDOOR* topology). As can be seen, for a network-wide monitoring infrastructure, the impact of monitoring overheads on data traffic can be substantial. However, by using a limited set of nodes, we can reduce the amount of overheads in the network. A similar impact was seen on the end-to-end delays of the data flows (shown in Fig. 2(b)). These results clearly indicate that using a network-wide monitoring infrastructure can generate large amounts of overhead, which can adversely impact the end users' performance. By using a smaller number of monitoring agents, we are able to alleviate the performance by a significant amount.

In order to further quantify the impact of monitoring overheads on WMNs, we also looked at the frame re-transmissions at the link layer. In 802.11 protocol, if a frame is lost, the link layer will re-transmit the frame a certain number of times, before reporting it as lost to the higher layers. If the number of re-transmissions is high, it will result in increased contention in the mesh network, and will also increase the end-to-end delay for the flow. Fig. 3 shows the improvement in link layer re-transmissions when only the nodes in the vertex-cover are used for monitoring, as compared to the network-wide monitoring. The reduction in the number of packets that are re-transmitted also explains the reduction in end-to-end delay in Fig. 2(b).

The results presented above confirm the fact that monitoring overheads can severely impact the performance of end users and that using a smaller number of monitors will reduce the communication overheads involved. However, it is also essential to maintain the accuracy of estimation of various parameters. By using a smaller number of monitors, we may have to sacrifice some "quality of information" in lieu of reduced monitoring overheads.

Table 1
Accuracy of delay estimation for monitor-selection approach.

Topology	No. of monitoring nodes	Delay from simulator (s)	Delay from measurements (s)	% Error
GRID	25	0.8	0.76	5
	13	0.54	0.49	9.26
OUTDOOR	15	0.62	0.595	4
	8	0.36	0.317	11.9
INDOOR	15	0.41 s	0.384	6.3
	9	0.263	0.23	12.5

Table 2
Performance of delay-based AODV for monitor-selection approach.

Topology	No. of monitoring nodes	Average throughput (Mbps)	Average delay (s)
GRID	25	1.16	0.72
	13	1.25	0.68
OUTDOOR	15	1.27	0.70
	8	1.42	0.62
INDOOR	15	1.03	1.17
	9	1.15	0.94

In order to investigate this, we decided to measure the end-to-end delays for various flows in our simulation. We first measured this value using all the nodes in the WMN as monitoring agents, and then using only the nodes in the vertex-cover set as monitoring agents. In both cases, we compared the "measured value" against that reported by the simulator. We obtain the "measured value" of end-to-end delay as follows. We measure the delay incurred by a packet at each hop along the route it takes to the gateway node. The sum of the per-hop delays gives us an estimate of the end-to-end delay. Table 1 shows the comparisons between the measured end-to-end delay, and the delay reported by the simulator, for each of the topologies. The client, for which the delay was estimated, was four hops away from the gateway node for all three topologies.

It was observed that reducing the number of monitoring agents caused the estimation error to approximately double for all the topologies. The reason for the measured values of delay to have larger errors when using a smaller number of monitors is as follows. When all the nodes in the network were being used for monitoring, each node could measure the delay involved in sending a packet to its neighbor. However, with the reduced number of monitors, for some links we are able to measure the delay in one direction only, and use this value as the delay in both the directions. Owing to the asymmetry of links in a wireless network, the delay on a link could be different in both the directions, which is captured in the first case, but not in the second. For example, for our *OUTDOOR* topology, the average delay from node 2 (DFG Hill) to node 10 (dan's repeater) was found to be 0.14 s, while that in the opposite direction was found to be 0.20 s. It was found the the link used a lower transmission rate in the opposite direction resulting in a larger delay.

The subsequent question that arises is whether the quality of information has degraded significantly or not. In other words, can such a measurement framework still be utilized for achieving a certain goal? In order to investigate this further, we decided to modify the existing implementation of AODV protocol in QualNet, to choose the next hop neighbors based on the delay values. That is, when AODV selects which next hop to choose to forward the packet to, the hop with the lowest delay is chosen. This mechanism can be used to provide delay-based QoS guarantees to the end users. We compare the performance of this modified AODV protocol for the two monitoring frameworks for all three topologies. Table 2 shows the average throughput and delay values measured from the network for the three topologies. *It can be seen that the proposed framework (with reduced monitoring agents) achieves better performance than the ideal case of network-wide monitoring.* This is because the increase in estimation error is offset by the reduction in overheads, resulting in improved network performance. Even though the protocol may not choose the best routes, there is less significantly less contention in the network, which improves the network performance. This reduced contention in the network is confirmed by the reduction in the number of link layer re-transmissions, which results in better performance at the higher layers. Hence, we can use the proposed monitoring framework for provisioning QoS in the network, while reducing overheads at the same time.

4.3. Reporting interval approach

In this section, we evaluate the impact of varying the frequency of reporting monitoring information. We investigate as to how this parameter impacts the measurement overheads and the accuracy of estimation. Reporting data very frequently (such as per-second basis) would provide us with a more accurate image of the network. However, such an approach may also incur large overheads. Fig. 4 shows the percentage of packets that were re-transmitted at the link layer for varying monitoring frequencies. As the reporting interval increases, there is less overhead in the network. As a result, a smaller number of packets needs to be re-transmitted at the link layer. The improvement in the number of re-transmissions results

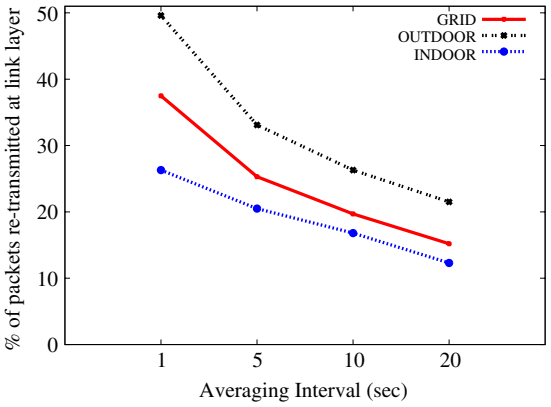


Fig. 4. % of packets re-transmitted at link layer.

Table 3
Accuracy of delay estimation with varying monitoring frequency.

Topology	Averaging interval	Delay from simulator (s)	Delay from measurements (s)	% Error
GRID	1	0.8	0.76	5
	10	0.68	0.60	11.76
OUTDOOR	1	0.46	0.43	6.52
	10	0.327	0.285	12.8
INDOOR	1	0.41	0.384	6.3
	10	0.32	0.278	13.12

Table 4
Performance of delay-based AODV with varying monitoring frequency.

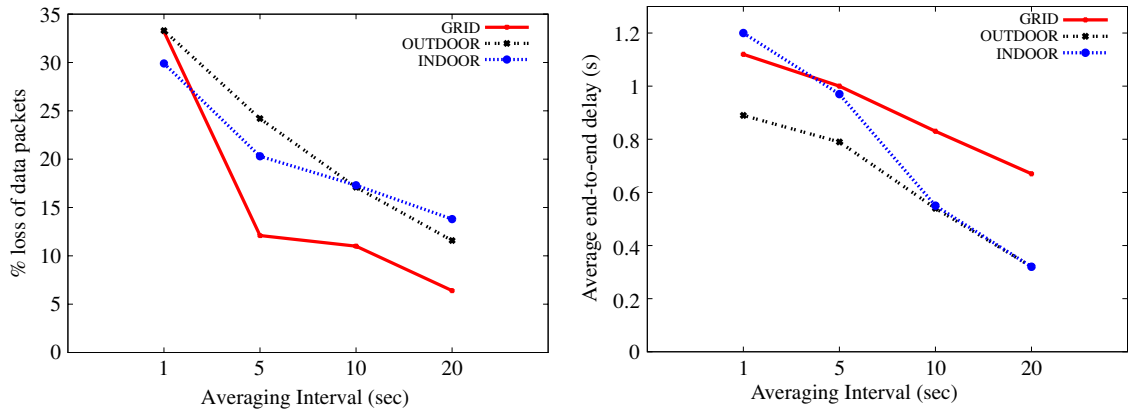
Topology	Averaging interval (s)	Average throughput (Mbps)	Average delay (s)
GRID	1	0.86	0.825
	10	1.09	0.64
OUTDOOR	1	0.97	0.485
	10	1.25	0.34
INDOOR	1	1.03	1.17
	10	1.24	0.95

in improved performance for end users (Fig. 5(a) and 5(b)). Since the amount of monitoring traffic reduces with decreasing frequency, there is less contention in the network, resulting in better performance for end users.

However, our objective is not to just reduce the volume of monitoring overheads in the network. We also want to maintain the functionality that needs to be achieved using the measurement data. We once again consider the example of delay-based routing. We evaluate the performance of our modified protocol for the two cases. In the first case, from every wireless node, we report network statistics to the central server every one second. That is, each node will send out one monitoring packet per second, destined for the central server. The second case is where each node sends out a measurement packet every ten seconds. In this case, every node maintains a simple moving average of the parameters that it is measuring and sends out a packet with the monitoring information after every ten seconds.

In order to analyze the performance of this scheme, we first compare the delay measurements using the two schemes. Table 3 shows the delay values for a particular client for the three topologies. When the monitoring information is sent every second, the accuracy is fairly high. When we send monitoring data averaged over ten seconds, the accuracy of estimation drops a little. This is expected because we have reduced the amount of monitoring packets being transmitted (reduced overheads) in the mesh network. This results in a reduction in the accuracy of estimating network statistics of interest (end-to-end delay in our case).

However, once again we wish to compare the actual performance of the application which is using the measurement data. We investigate the efficiency of this approach by comparing the performance of our delay-based routing protocol for both the scenarios. Table 4 compares the results for the three topologies. As can be seen, averaging the data over ten seconds, achieves slightly better performance than the first case, owing to the reduction in measurement overheads. These results indicate that we can easily use a lower frequency of reporting monitoring data for QoS provisioning, without sacrificing our performance.



(a) Packet loss.

(b) End-to-end delay.

Fig. 5. Impact of monitoring frequency on user performance.

Table 5
Impact of threshold-monitoring approach on % loss of data packets.

Topology	Reporting mechanism	% loss of data packets
GRID	Periodic Threshold	21.6 5.86
OUTDOOR	Periodic Threshold	27.26 14.69
INDOOR	Periodic Threshold	29.9 11.6

Table 6
Impact of threshold-monitoring approach on end-to-end delay.

Topology	Reporting mechanism	Avg. end-to-end delay (s)
GRID	Periodic Threshold	1.15 0.82
OUTDOOR	Periodic Threshold	0.58 0.375
INDOOR	Periodic Threshold	1.32 0.61

4.4. Threshold-based monitoring

In this section, we present the results for the threshold-based monitoring approach. As explained earlier, such a scheme would be highly beneficial for tracking anomalies in network performance. We wish to evaluate the trade-offs involved, in terms of overheads and accuracy, between the two choices of using periodic monitoring and threshold-based monitoring. We consider the specific example of tracking the traffic load on each node as an indication of the network being congested. We wish to identify the events when the traffic load on a node crosses a certain pre-defined threshold, and report it to the central controller. The threshold was set to be one standard deviation above the average congestion observed in our simulations. For every packet generated by a node, the node keeps track of the time at which the packet was put in the output queue, and the time at which the link layer acknowledgement for the packet was received. The difference in these two time intervals gives us an estimate of the congestion at the node. If the network is congested, neighboring transmissions may cause the packet to stay in the output queue for a longer time. It may also happen that the transmitted frame is lost (or ACK timeout occurs), and the frame is re-transmitted. Both these scenarios will increase the time interval of the ACK reception, indicating that the network is congested. Each node keeps track of this parameter and reports it to the central controller. The metric is reported periodically for the periodic framework, and only if it crosses a certain threshold, for the threshold-based framework.

Both periodic and threshold-based monitoring can be used to achieve this objective. However, it was observed that periodic reporting can lead to large delays and packet losses for end users. Table 5 shows the percentage loss for data packets for the three topologies, while Table 6 shows the average end-to-end delay. By using threshold-based monitoring, we can reduce the amount of overhead in the network, thereby improving the performance for the data flows (as indicated by lower packet losses and delay).

Table 7
Impact of threshold-monitoring approach on measurement delivery ratio.

Topology	Periodic monitoring	Threshold-based monitoring
GRID	0.213	0.647
OUTDOOR	0.369	0.736
INDOOR	0.265	0.936

We further investigated whether the threshold-based scheme satisfies our second objective of maintaining the desired application performance. We use both the periodic and the threshold-based monitoring mechanisms to report data to the central server. In the periodic framework, one monitoring packet per second is sent to the central server, irrespective of the load on the mesh node. The central server will check the data in the received packet and determine if the reporting node is congested. In the threshold-based mechanism, each node in the mesh network will send a monitoring packet only if it perceives itself as being congested, based on the criterion explained above. This helps us in reducing the volume of monitoring traffic in the mesh network. At the same time, we see an improvement in the performance of the monitoring framework. Table 7 shows the ratio of the "total *useful* monitoring packets" received at the central server to the "total monitoring packets" sent out by all the nodes in the mesh network. We term this ratio as the *measurement delivery ratio*. We define the "total *useful* monitoring packets" as the number of monitoring packets that help us identify node congestion (that is, the reported value of traffic load on the node is above the threshold). *A greater measurement delivery ratio translates to more useful information being transmitted per monitoring packet, and hence indicates a better utilization of the network resources.* The measurement delivery ratio is not one hundred percent for the threshold-based approach as well, because some monitoring packets may be lost in the network due to collisions. The periodic framework suffers from collisions, as well as transmission of redundant information.

It was observed that with the periodic monitoring framework, a lot of monitoring packets were lost. Monitoring packets that report traffic load below the threshold, do not actually provide us with useful information, while consuming network resources. These unnecessary transmissions add to the contention in the network, and as a result, there is loss of information at the central server, due to which we may not be able to accurately identify as node congestion. On the other hand, with the threshold-based monitoring framework, the loss of monitoring packets is much less, and the central server has more accurate and up-to-date information of the traffic load on various nodes. As a result, even though fewer monitoring packets are being transmitted, the measurement delivery ratio is higher. This is because fewer packets are lost and a higher percentage of monitoring packets reaches the central server. Moreover, all these packets contain *useful information*. Hence, the threshold-based scheme will be able to predict anomalies in the network performance (increased congestion in our case) with greater accuracy.

4.5. Inferences from simulation studies

Based on the results from our simulation-based studies, we had the following two primary inferences. First, even low amounts of monitoring overhead can adversely impact transmission of data traffic. This was clearly visible in our results, where even small-size and low-rate management traffic clearly impacted the end-user's performance. Second, we learned that it is feasible to maintain the quality of information and performance of application above a threshold, while reducing overheads in the network. We observed that even by using lower number of monitoring agents, or by using threshold-based monitoring, we are still able to achieve the desired application-layer objective.

However, as has been shown in numerous previous works [35,36], simulation studies do not always provide the most accurate picture, especially for the case of wireless networks. Present day simulators rely on simple underlying physical models that do not capture the real world propagation effects. They also make simplifying assumptions regarding other features of a wireless network, such as communication and interference ranges and so on. Hence, in order to confirm our findings, we further explore this issue by performing extensive experiments on our laboratory testbed.

5. Experimental evaluation

In order to re-affirm our simulation results, we carry out several experiments using our laboratory testbed. We investigate two of the proposed approaches: the monitor selection approach (changing the number of monitoring agents in the network) and the reporting interval approach (changing the frequency of reporting monitoring data). By varying these parameters, we investigate the impact of monitoring overheads on data traffic. We further evaluate the impact of varying these parameters on the accuracy of measurement data and hence study the trade-offs involved between monitoring accuracy and overheads. Before presenting the results of our experimental evaluation, we briefly describe the functioning of our monitoring tool and our experimental testbed.

5.1. Monitoring tool and testbed

We have developed a monitoring tool for wireless networks that has been implemented in our testbed. Our tool consists of a client and a server side. The client resides on the mesh nodes being used as monitoring agents. The monitoring client

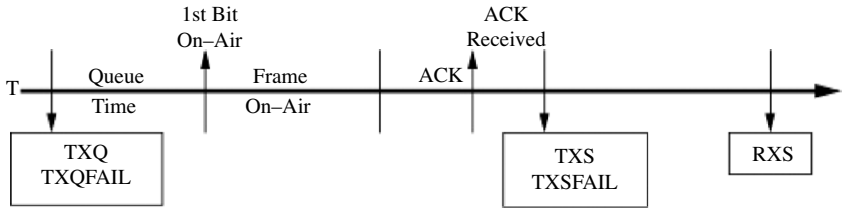


Fig. 6. 802.11 frame events.

collects specified link layer statistics and sends them to the monitoring server. The monitoring server could reside with the gateway node or on a separate dedicated infrastructure. It receives information from the various monitoring agents and stores them for analysis. We implemented the framework in the Linux kernel, using the madwifi-ng wireless device driver. We modified this device driver to report certain 802.11 events, from which we can derive information about the wireless channel. The user-space client program residing on the mesh nodes communicates with the modified driver through the NetLink [30] library and records these events. The user-space program can then process these events, to obtain the necessary information. Fig. 6 shows the temporal relationship between the various 802.11 events reported by the modified driver. Three types of events are reported, each of which corresponds to an action taken by the driver in response to a frame transmission or reception. These three events are:

- Frame enqueue (TXQ/TXQFAIL) event is generated when the device driver receives a frame from the higher layers. The frame is either placed in the output queue of the wireless interface (TXQ), or dropped because the queue is full (TXQFAIL).
- Frame transmission status (TXS/TXSFAIL) occurs when either the link layer acknowledgement (ACK) for a transmitted packet is received (TXS), or the ACK is not received and the timer expires (TXSFAIL).
- Frame receive status (RXS) event is generated whenever a MAC frame or an ACK is received.

The reader can refer to [31] for further details on the monitoring tool. Our mesh network testbed consists of ten nodes deployed around our laboratory. These nodes consist of HP nc6000 laptops and Soekris net4826 embedded devices. Both types of nodes run a customized version of Linux and use the madwifi-ng wireless driver. One of the laptops serves as the gateway node, with most nodes being two to three wireless hops away. The gateway node also serves as the central server for collecting monitoring data. All the mesh nodes involved in network monitoring transmit data to the gateway node. The nodes are configured in 802.11a mode, with the auto-rate feature enabled. No other 802.11a networks were detected in the vicinity of our testbed.

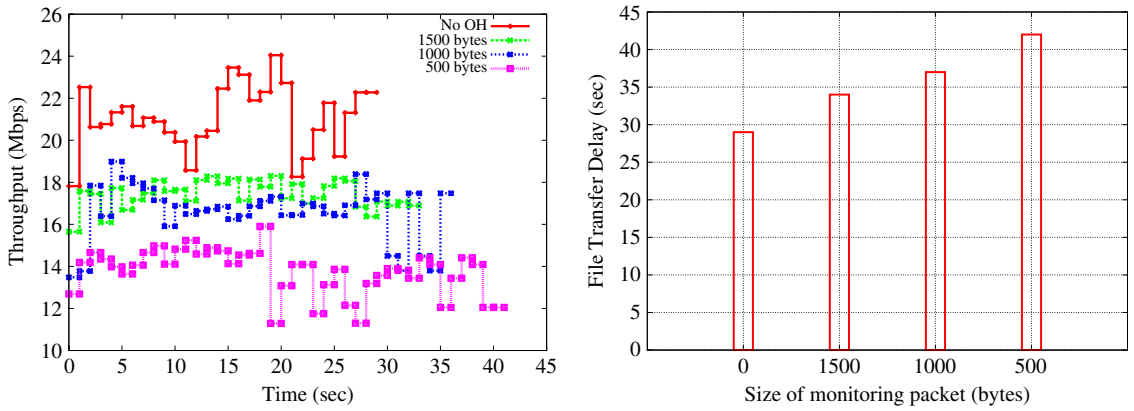
5.2. Impact of size of monitoring packet

In this section, we present the results for the performance of the reporting interval approach on our testbed. For each data packet, our tool collects a fixed amount of information (such as packet length, modulation rate at which it was sent and so on). Once the specified amount of data is collected, the monitoring client will create a new monitoring packet and transmit it to the monitoring server. The amount of information that is collected each time before a monitoring packet is sent out can be defined by the user. For example, for a given traffic load, if we have the choice of using 1500 byte monitoring packets, and 500 byte monitoring packets, then a larger number of monitoring packets will be sent out in the latter case. This is because the smaller packets will be filled up with monitoring information more quickly. Hence, for a given traffic load, we are able to control the frequency of reporting measurement data, by controlling the size of the packet used for sending the monitoring data.

Using a smaller packet size (higher frequency of reporting data) will cause more monitoring packets to be sent out from each mesh node. This will introduce large amounts of overhead in the wireless network. However, the advantage of using a higher frequency of reporting data is that measurement information will be sent out more quickly from the monitoring agents to the central server, thereby providing us with a more accurate image of the network status. On the other hand, if we use a larger packet size (smaller frequency of reporting data), then the mesh node will collect measurement data for a longer period of time before actually sending out the monitoring packet. Hence, there will be some delay between the time when the data is collected at a mesh node, and the time when it is actually transmitted and reaches the server. Thus there is a trade-off involved, in terms of the overheads introduced in the network, and the quality of monitoring information. We investigated this issue using our laboratory testbed. For each run, we changed the size of monitoring packets. We used three sizes of 1500 bytes, 1000 bytes, and 500 bytes.

5.2.1. Impact of overheads

In order to evaluate the impact of monitoring overheads on TCP flows, we started a file transfer between two nodes. We first performed several runs of the file transfer at different points in time, without any other transmissions. Each time, the file transfer took 29 s to completion. This was done to establish the base performance of the FTP transfer in the absence of



(a) Impact on throughput.

(b) Impact on FTP delay.

Fig. 7. Impact of monitoring packet size on user performance for FTP data.

Table 8
Impact of monitoring packet size on link layer re-transmissions for FTP.

Packet size	% of data packets re-transmitted
No OH	1.45
1500 bytes	4.25
1000 bytes	5.6
500 bytes	7.75

Table 9
Impact of monitoring packet size on link layer re-transmissions for UDP.

Packet size	% of data packets re-transmitted
No OH	1.23
1500 bytes	5.37
1000 bytes	8.6
500 bytes	12

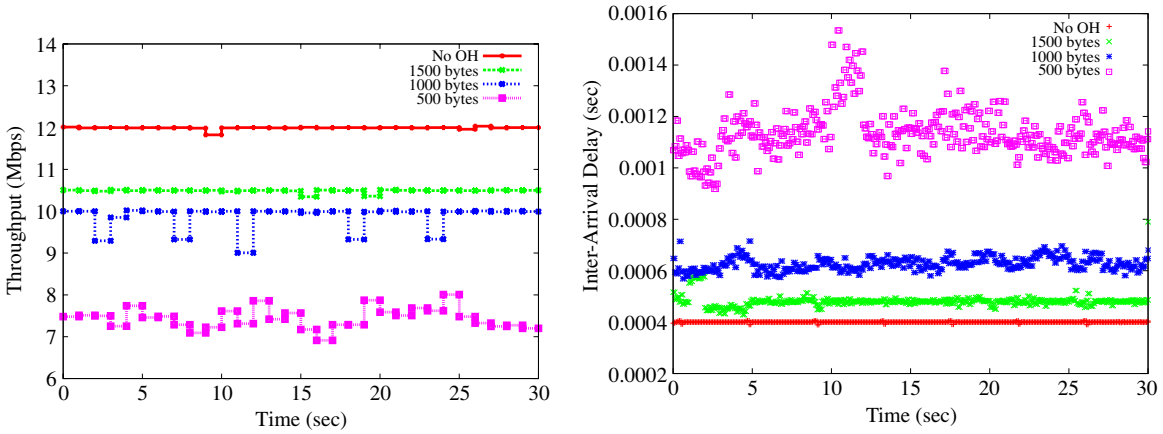
any interference. We then enabled the monitoring agents on eight of the ten nodes and evaluated the performance of the FTP transfer, while varying the size of monitoring packets.

Fig. 7(a) shows how the throughput of the FTP flow varies with changing packet size. As the size of monitoring packet decreases (reporting frequency increases), more overheads are introduced into the network, thereby adversely impacting the throughput of the FTP flow. Fig. 7(b) shows the amount of time it took to transfer the file in each case. Once again, the delay in transferring the file shows an increasing trend, with an increase in monitoring overheads.

We further investigated re-transmissions at the MAC layer. We tracked the number of packets that were transmitted more than once at the link layer. It has been shown in previous works that in 802.11 networks, most packets are transmitted successfully in the first attempt. Any extra re-transmissions add to the contention in the network. This increases the end-to-end delay of the flow, resulting in degraded performance. Table 8 shows the percentage of data packets that were re-transmitted more than once at the link layer.

In order to evaluate the impact of monitoring overheads on UDP traffic, we used a traffic generator to exchange UDP data between two nodes. Once again, we first performed several runs at different points in time, without any other transmissions on the network. This was done to establish the base performance of the UDP transfer, in the absence of any interference. The maximum UDP rate that our testbed could support, without any packet loss, was found to be 25 Mbps. We introduced UDP traffic at a much lower rate of 12 Mbps, in order to avoid saturating the channel. We then enabled the monitoring agents on the eight nodes, and again evaluated the performance of the UDP transfer, while varying the size of monitoring packets.

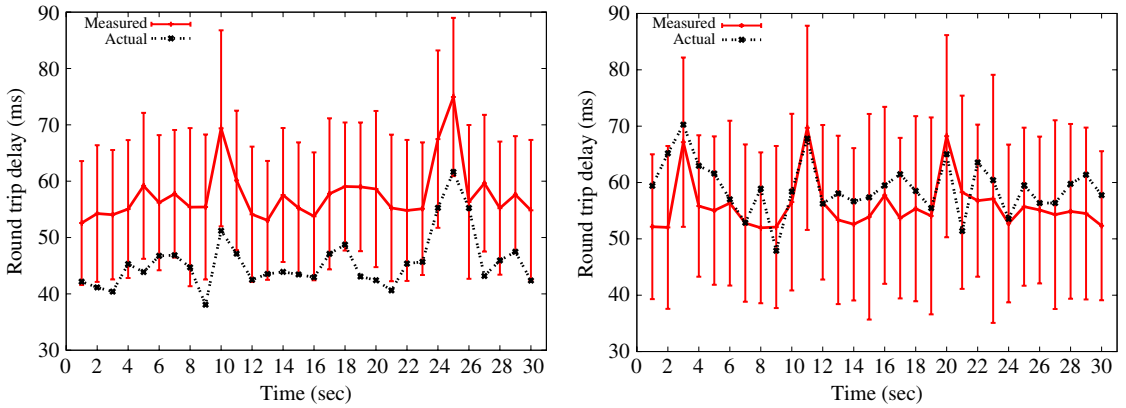
Fig. 8(a) shows how the throughput of the UDP flow varies with changing packet size. Similar to the FTP scenario, more measurement overhead translates to poorer performance for the end user. Fig. 8(b) shows the variation in the inter-arrival delay between the UDP packets. These packets are sent at a constant rate from the source. However, from the figure we can see that as the frequency of reporting monitoring data goes up (decreasing packet size), the inter-arrival time shows an increasing trend. This is because of the monitoring overheads in the network. As the amount of overhead increases, the UDP data packets face increased contention and delay in the network. Similar to the TCP scenario, we also looked at frame re-transmissions at the MAC layer. Table 9 shows the percentage of data packets that were re-transmitted more than once at the link layer. With increasing amounts of overhead in the network, there is more contention and hence more and more data packets have to be re-transmitted.



(a) Impact on throughput.

(b) Impact on packet inter-arrival time.

Fig. 8. Impact of monitoring packet size on user performance for UDP data.



(a) 1500 byte monitoring packet size.

(b) 500 byte monitoring packet size.

Fig. 9. Accuracy of delay measurement for FTP traffic with varying monitoring packet size. 'Actual' refers to the IPerf delay, while 'Measured' is the delay obtained from our monitoring tool.

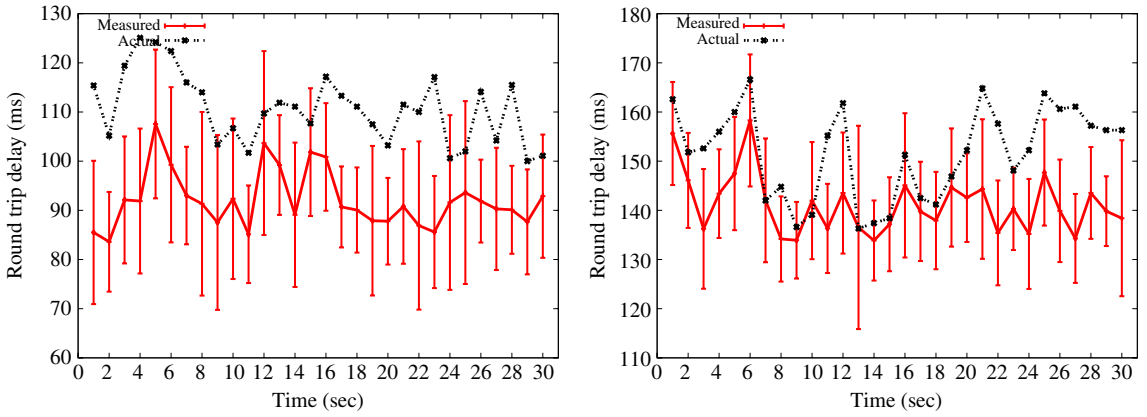
5.2.2. Accuracy of estimation

As was seen in the previous section, reducing the size of monitoring packets results in more measurement frames being sent out per unit time. This increases the contention in the network, thereby impacting the performance of end users. However, at the same time, using smaller packets will result in the measurement information being sent out more quickly to the central server. This would help us in maintaining a more accurate and up-to-date image of the network status.

In order to experimentally verify this, we compare the round-trip delay measurements obtained from a traffic generator (IPerf), and those obtained via our monitoring framework. We used IPerf to transmit data between two nodes for thirty seconds. IPerf reports the throughput and round-trip delay between the nodes. We compared this delay against that obtained from our monitoring framework. Fig. 9 shows the variations in the delay measurement for FTP traffic, with varying packet sizes. As can be seen, with smaller packet size (larger reporting frequency), the measured delay estimates are much closer to the actual delay values. A similar trend was observed for UDP traffic (Fig. 10). Fig. 11 shows the CDF of the percentage error between the "actual" value measured using the traffic generator used, and the average "measured" value obtained from our monitoring tool.

5.3. Impact of number of monitors

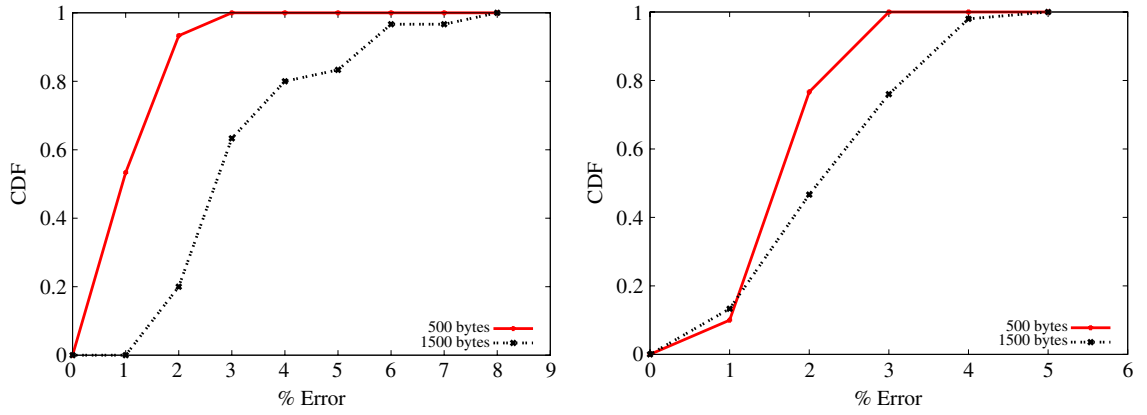
Using simulations, we have demonstrated the efficiency of using the vertex-cover technique to identify the nodes that can be used for monitoring the mesh network. This helps us in reducing the monitoring overheads in the network, as compared to a framework that spans the entire network. Our simulations also showed that this approach is effective in maintaining the quality of information needed by the higher layer applications. We further decided to investigate the efficiency of this approach via experiments. For our ten node testbed, the vertex-cover consists of four nodes only. We first evaluate the



(a) 1500 byte monitoring packet size.

(b) 500 byte monitoring packet size.

Fig. 10. Accuracy of delay measurement for UDP traffic with varying monitoring packet size.



(a) FTP traffic.

(b) UDP traffic.

Fig. 11. CDF of percentage error in estimation of round-trip-delay with varying monitoring packet size.

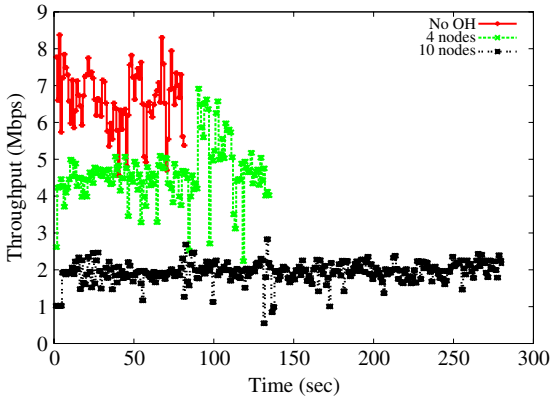
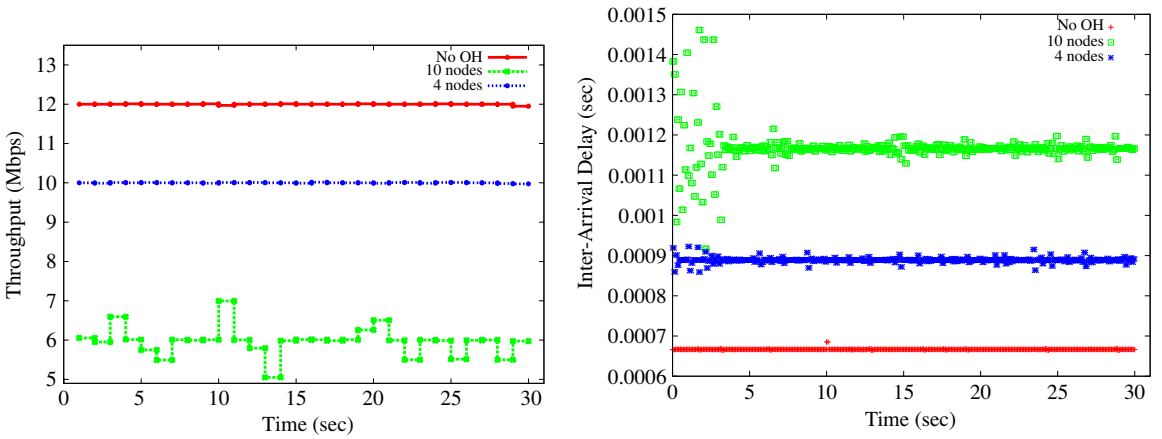


Fig. 12. Impact of number of monitoring nodes on FTP throughput.

network performance with all the nodes in the network being used as monitoring agents. Subsequently, we use only the four nodes in the vertex-cover set as monitoring agents, and compare the trade-offs for overheads and accuracy of measurement.

5.3.1. Impact of overheads

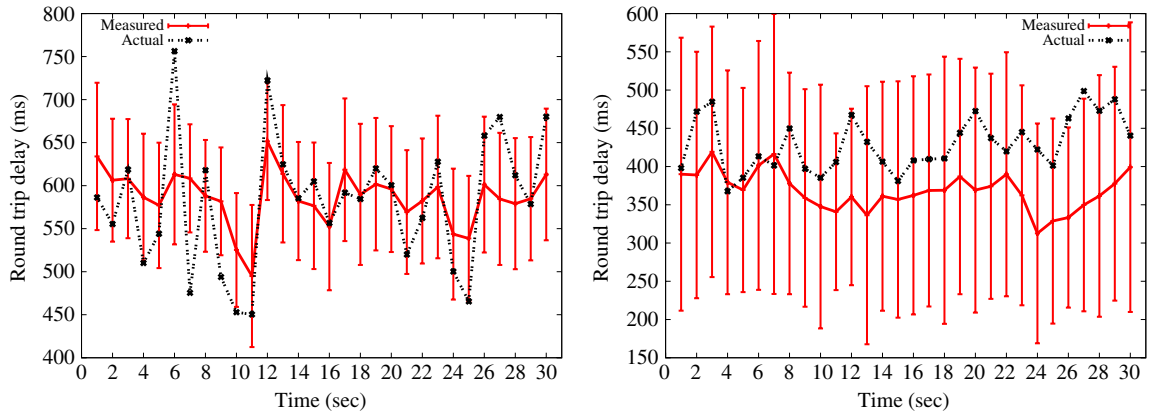
Similar to the previous section, we started a FTP file transfer between two clients. We first measured the performance of the file transfer without any other transmissions in the network. Subsequently, we repeated the experiment, first with



(a) Impact on throughput.

(b) Impact on packet inter-arrival time.

Fig. 13. Impact of number of monitoring nodes on user performance for UDP data.



(a) 10 monitoring agents.

(b) 4 monitoring agents.

Fig. 14. Accuracy of delay measurement for FTP traffic with varying number of monitoring agents.

Table 10
Impact of number of monitoring nodes on FTP delay.

No. of monitoring nodes	FTP delay (s)
No OH	82
10	280
4	135

all ten nodes monitoring the network, and then with only the four vertex-cover-set nodes monitoring the network. Fig. 12 shows the variation in the throughput of the FTP flow for each case. Table 10 shows the total time it took to complete the file transfer between the two clients. We also tracked the number of packets that were transmitted more than once at the link layer. These re-transmissions increase the end-to-end delay of the flow, resulting in poorer performance. Table 11 shows the percentage of data packets that were re-transmitted multiple times at the link layer.

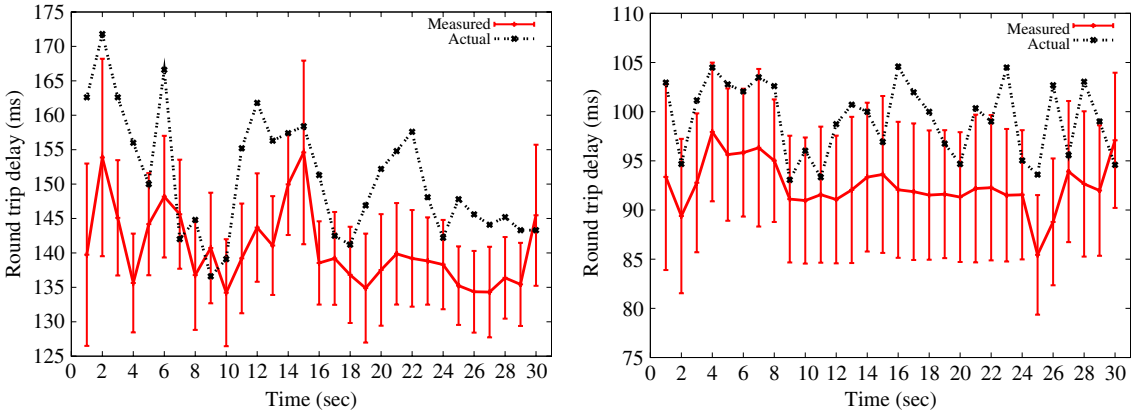
We also evaluated the impact of number of monitoring nodes on UDP flows. We introduced UDP traffic at a constant rate of 12 Mbps between the source and the destination. We then evaluated the performance with all the ten mesh nodes being used as monitoring agents, and only the four nodes in the vertex cover set being used as monitoring agents. Fig. 13(a) shows the variation in the throughput of the UDP flow, with different number of monitoring agents. Similar to the FTP scenario, more measurement overhead translates to poorer performance for the end user. Fig. 13(b) shows the variation in the inter-arrival delay between the UDP packets. From the figure we can see that when all the mesh nodes are used as monitoring agents, the inter-arrival time shows an increasing trend. Table 12 shows the impact of the number of monitoring agents on the 802.11 re-transmissions. More overheads translate to larger packet losses and more MAC layer re-transmissions, further degrading the network performance.

Table 11
Impact of number of monitoring nodes on link layer re-transmissions for FTP.

No. of monitoring nodes	% of data packets re-transmitted
No OH	1.15
10	4.56
4	2

Table 12
Impact of number of monitoring nodes on link layer re-transmissions for UDP.

No. of monitoring nodes	% of data packets re-transmitted
No OH	1.06
10	4.23
4	1.97



(a) 10 monitoring agents.

(b) 4 monitoring agents.

Fig. 15. Accuracy of delay measurement for UDP traffic with varying number of monitoring agents.

5.3.2. Accuracy of estimation

As was seen in the previous section, reducing the number of monitoring agents helps us in reducing the monitoring overheads, while still being able to monitor every link in the network. However, as explained in the previous section, using a smaller number of monitors may impact the quality of information being monitored. For example, we may not be able to monitor some links in both the directions, and owing to the link-asymmetry in multi-hop wireless networks, this would adversely impact the accuracy of measurement data. In order to experimentally investigate this, we compare the round-trip delay measurements obtained from the traffic generator IPerf and those obtained via our monitoring framework. Fig. 14 shows the variations in the delay measurement for FTP traffic, with varying number of monitoring agents. Similar results were obtained with UDP type traffic (Fig. 15). Fig. 16 shows the CDF of the percentage error between the "measured" values and the "actual" delay values.

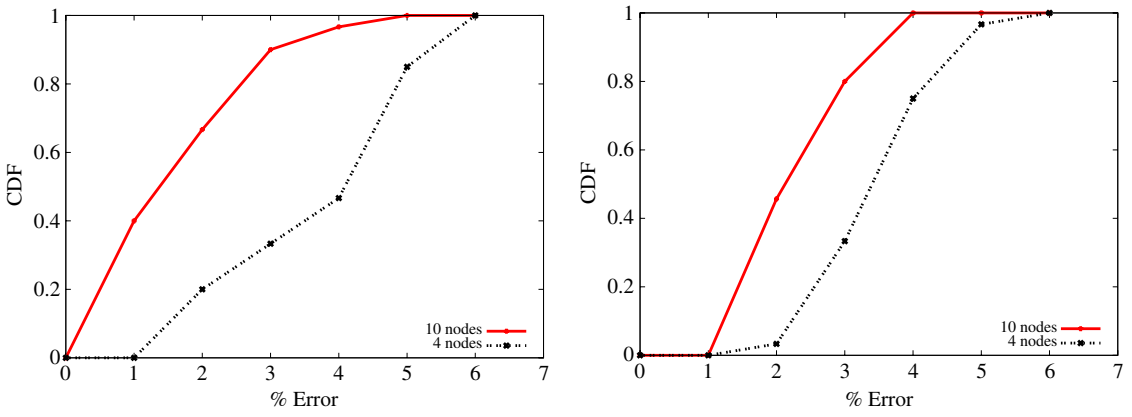
Ideally, it would seem that the accuracy of delay measurements should go down when we use only the vertex-cover nodes for monitoring the network. However, we observed that there is not much reduction in accuracy. This is important because by using a reduced number of nodes as monitoring agents, we were able to reduce the overheads in the network, and use more network capacity for transmitting user data. At the same time, our accuracy of estimation is fairly good, as compared to the ideal case where every mesh node is used as a monitoring agent. This means that we can use this new framework for collecting network statistics, and use that data for a higher layer application such as QoS routing or admission control.

5.4. Inferences from experimental evaluation

We experimented with both FTP and uniform CBR traffic types on our laboratory testbed. We observed that for both traffic types, management overheads severely impact the user performance. We also observed how different techniques can help us reduce these overheads, while still maintaining a high level of accuracy of estimating network parameters. This serves to re-affirm our simulation results, and shows how monitoring techniques can be made more efficient.

6. Conclusions, inferences and future work

In this paper, we have looked at the issue of efficient monitoring in wireless mesh networks. With their growing popularity and increasing applications, several schemes for implementing Quality of Service and developing measurement-



(a) FTP traffic. (b) UDP traffic.

Fig. 16. CDF of percentage error in estimation of round-trip-delay with varying number of monitoring agents.

based models for wireless mesh networks have been proposed. Most of these schemes rely on an underlying monitoring framework, which collects the necessary statistics from the wireless network. However, *the impact of monitoring overheads on the transmission of data traffic in wireless networks has not been studied so far.* Most previous works that propose active-measurement-based schemes for routing or fault management in wireless mesh networks, have overlooked the issue of overheads. Thus, we first look at the impact of monitoring traffic on the forwarding of user data traffic, for different applications. Via extensive simulations and experiments, we evaluate the performance of several schemes for reducing the monitoring overheads in WMNs. We look at monitor selection based on network characteristics such as topology, changing frequency of reporting monitoring data, and threshold-based monitoring, as possible solutions to the problem of reducing overheads. We evaluate as to how these schemes lead to an improvement for the end users' performance, in terms of packet loss and delay. We also investigate whether these techniques impact the desired functionality for which the network is being monitored. We evaluate the performance of different applications using these monitoring techniques. Some of the important lessons learned as part of our work are:

- Given the importance of measurement-based approaches for providing Quality of Service and fault management in wireless mesh networks, it is crucial to study the impact of monitoring traffic on the user data traffic. Through our study, we find that periodic monitoring of a network can cause data loss of as much as 40% and can severely impact the network performance from an end user's perspective.
- By using different techniques such as constrained number of monitors and threshold-based monitoring, we can greatly improve the network performance. These techniques help us in maintaining the desired level of measurement accuracy, while reducing the associated overheads.
- We observed that different monitoring techniques lend themselves to different application scenarios. It is crucial to use the right technique for an application, in order to maintain the balance between reduction in overheads and accuracy of measurement data.

As part of our future work, we intend to study the impact of the proposed schemes on more varied topologies. We also plan to see what different metrics can we measure using such frameworks and with what accuracy. We would also like to investigate further applications of these monitoring frameworks. A monitoring framework that can adapt itself to the network status is also an objective of our future research. Furthermore, we plan to study the case where multiple such schemes may need to be combined together in order to serve several QoS applications simultaneously.

References

[1] P. De, A. Raniwala, S. Sharma, T. Chiueh, Design considerations for a multihop wireless network testbed, Communications Magazine, IEEE 43 (2005).
[2] D. Wu, D. Gupta, S. Liese, P. Mohapatra, Qurinet: quail ridge natural reserve wireless mesh network, in: 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, WiNTECH, 2006.
[3] J. Bicket, D. Aguayo, S. Biswas, R. Morris, Architecture and evaluation of an unplanned 802.11b mesh network, in: 11th Annual International Conference on Mobile Computing and Networking, MobiCom, 2005.
[4] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: 10th annual International Conference on Mobile computing and Networking, MobiCom, 2004.
[5] A. Raniwala, T. Chiueh, Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network, in: 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 2005.
[6] M. Kodialam, T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, MobiCom, 2005.
[7] D. Gupta, D. Wu, C. Chen, P. Mohapatra, C. Chuah, S. Rungta, Experimental study of measurement-based admission control for wireless mesh networks, in: 4th EEE International Conference on Mobile, Ad-hoc and Sensor Systems, MASS, 2007.
[8] L. Chen, W.B. Henzelman, Qos-aware routing based on bandwidth estimation for mobile ad hoc networks, Selected Areas in Communications IEEE Journal 23 (2005) 561–572.

[9] H. Zhu, I. Chlamtac, Admission control and bandwidth reservation in multi-hop ad hoc networks, in: Computer Networks, 2005.
[10] I.D. Chakeres, E.M. Belding-Royer, Pac:perceptive admission control for mobile wireless networks, in: 1st International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, QSHINE, 2004.
[11] D. DeCouto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: 9th Annual International Conference on Mobile Computing and Networking, MobiCom, 2003.
[12] R. Chandra, V. Padmanabhan, M. Zhang, Wifiprofiler: cooperative diagnosis in wireless lans, in: 4th international Conference on Mobile Systems, Applications and Services, MobiSys, 2006.
[13] P. Bahl, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, B. Zill, Dair: A framework for managing enterprise wireless networks using desktop infrastructure, in: 4th Workshop on Hot Topics in Networking, HotNets, 2005.
[14] Y. Cheng, J. Bellardo, P. Benkö, A. Snoeren, G. Voelker, S. Savage, Jigsaw: solving the puzzle of enterprise 802.11 analysis, in: SIGCOMM Computer Communication Review, 2006.
[15] D. Gupta, P. Mohapatra, C.-N. Chuah, Efficient monitoring in wireless mesh networks: Overheads and accuracy trade-offs, in: 5th IEEE International Conference on Mobile, Ad-hoc and Sensor Systems, MASS, 2008.
[16] P. Moghe, M. Evangelista, An adaptive polling algorithm, in: IEEE/IFIT Network Operations and Management Symposium, NOMS, 1998.
[17] K. Yoshihara, K. Sugiyama, H. Horiuch, S. Obana, Dynamic polling scheme based on time variation of network management information values, in: Integrated Network Management Symposium, IM, 1999.
[18] J. Jiao, S. Naqvi, B. Sugla, Towards efficient monitoring, Selected Areas in Communications, IEEE Journal 18 (2000).
[19] M. Dilman, D. Raz, Efficient reactive monitoring, Selected Areas in Communications, IEEE Journal 20 (2002).
[20] S. Kyoungwon, G. Yang, J. Kurose, D. Towsley, Locating network monitors: complexity, heuristics, and coverage, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 2005.
[21] G. Cantieni, G. Iannaccone, C. Barakat, C. Diot, P. Thiran, Reformulating the monitor placement problem: Optimal network-wide sampling, in: 40th Annual Conference on Information Sciences and Systems, 2006.
[22] J. Horton, A. López-Ortiz, On the number of distributed measurement points for network tomography, in: 3rd ACM SIGCOMM Conference on Internet Measurement, IMC, 2003.
[23] C. Chaudet, E. Fleury, I. Lassous, H. Rivano, M. Voge, Optimal positioning of active and passive monitoring device, in: ACM Conference on Emerging Network Experiment and Technology, CoNEXT, 2005.
[24] X. Liu, J. Yin, Z. Cai, S. Lv, On the placement of active monitor in ip network, in: 3rd International Conference on Computer Networks and Mobile Computing, ICCNMC, 2005.
[25] B. Keegan, K. Kowalik, M. Davis, Experimental measurement of overhead associated with active probing of wireless mesh networks, in: IEEE International Conference on Signal Processing and Communications, ICSPC, 2007.
[26] J. Camp, V. Mancuso, O. Gurewitz, E. Knightly, A measurement study of multiplicative overhead effects in wireless networks, in: 27th IEEE International Conference on Computer Communications, INFOCOM, 2008.
[27] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2001.
[28] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, J. Zahorjan, Measurement-based models of delivery and interference in static wireless networks, in: SIGCOMM Computer and Communications Review, 2006.
[29] C. Perkins, E.M. Belding-Royer, Ad-hoc on-demand distance vector routing, in: 2nd IEEE Workshop on Mobile Computer Systems and Applications, 1999.
[30] C. Benvenuti, Understanding Linux Network Internals, O' Reilly, 2006.
[31] D. Wu, P. Djukic, P. Mohapatra, Determining 802.11 link quality with passive measurements, in: IEEE International Symposium on Wireless Communication Systems, 2008.
[32] S. Chen, K. nahrstedt, Distributed QoS routing with imprecise state information, in: 7th International Conference on Computer Communications and Networks Systems, 2008.
[33] Q. Han, N. venkatasubramaniam, Information Collection Services for QoS-aware Mobile Applications, in: IEEE Transactions on Mobile Computing, 2006.
[34] A. Shaikh, J. Rexford, K.G. Shin, Evaluating the impact of stale link state on QoS routing, IEEE/ACM Transactions on Networking (2001).
[35] D. Johnson, Validation of wireless and mobile network models and simulation, in: DARPA/NIST Network Simulation Validation Workshop, 1999.
[36] C. Newport, D. Kotz, Y. Yuan, R.S. Gray, J. Liu, C. Elliott, Experimental evaluation of wireless simulation assumptions, in: 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2004.