Poster Abstract: A New Scheme on Link Quality Prediction and its Applications to Metric-Based Routing

Yong Wang, Margaret Martonosi, and Li-Shiuan Peh
Princeton University
{yongwang,mrm,peh}@princeton.edu

ABSTRACT

Efficient routing in wireless sensor networks entails the establishment of high quality links. Recent research has shown that metric-based routing, such as ETX [6], can significantly improve routing performance by tracking various link-quality metrics. Such metrics, however, may fail to capture link quality at relatively high traffic rates. This poster describes how machine learning techniques can be leveraged to help estimate link quality in those adverse scenarios. We also present MetricMap, a metric-based protocol using our learning-enabled link quality assessment method. Experimental results on MoteLab show that MetricMap can achieve up to 400% improvement on data delivery rate in a high traffic rate application, with no negative impact on other performance metrics, such as data latency.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Sensor networks, Link quality, Supervised learning, Classification

1. INTRODUCTION

Due to the unreliable, asymmetrical and unpredictable characteristics of radio transmissions in many sensor network applications, metric-based routing protocols that track link quality and select higher-quality links over lossy ones have been studied and shown to perform well in real-world sensor networks, when wireless channel conditions vary significantly over time and space [6, 1].

The utility of metric-based routing heavily depends on the accuracy of link quality measurements. When the sensor sampling rate increases, traditional metric-based routing protocols cannot handle the increase in traffic anymore. The root cause for such failures is that metric-based routings rely on monitoring data traffic or probe packets to derive the probability of transmission success over a certain link. However, a simple experiment on MoteLab using the `MintRoute` component provided by TinyOS shows that such a measurement mechanism fails to capture the underlying link quality at a traffic rate of 2 packets/second (pps) or higher. As a result, these derived link-quality metrics do not provide useful information. Either the forward transmission probability or the backward transmission probability of a link derived this way arrives at a value indicating barely any transmissions can be carried through this link. Under such circumstances, the MT protocol [6], which uses the `MintRoute` component, cannot find the route to the sink. However, because not all links are overloaded, the routing process can be resumed once an accurate estimation of link quality is in place.

We propose to use supervised machine learning algorithms to classify link qualities. In particular, we use data that has been pre-classified offline based on link quality indicator (LQI) provided by 802.15.4 radios (such as CC2420 in the MicaZ series of motes) to one of a number of categories. Sets of data consisting of the pre-assigned category combined with features of nodes (such as node depth, buffer size, packet RSSI value) are used to train the classifier.

2. METHODOLOGY

In this section, we briefly introduce supervised learning and discuss the infrastructure we used for link quality learning.

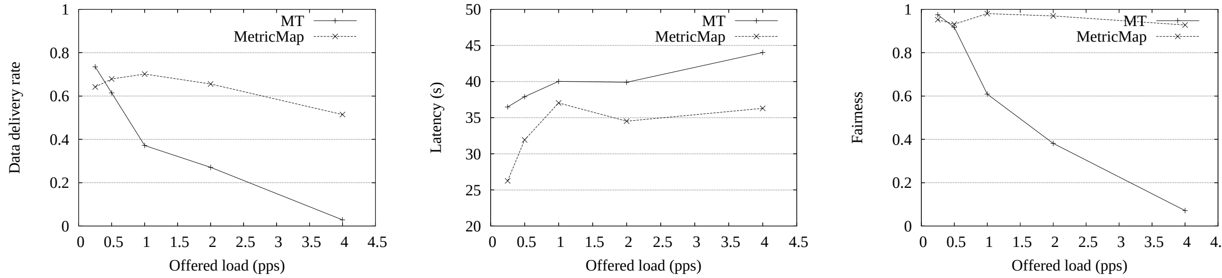
2.1 Supervised Learning

The goal of supervised learning is to predict the value of an outcome measure based on a number of input measures [3]. The outcome measure could be numerical or categorical. Learning is performed on a set of training samples. Each sample $< x_i, y_i >$ consists of a feature vector x_i and a corresponding class label or numerical value y_i. The feature vector contains measurable features of the system under consideration. If the outcome is categorical, the learning becomes a classification problem. Training a classifier usually involves finding a mapping from feature vectors to output labels so that the overall classification error is minimized on the training samples. A good learner should be one that accurately predicts new samples not in the training set. We evaluated two classifiers — decision trees and classification rules.

Decision trees. Decision tree algorithms are widely used in solving classification problems. They take a “divide-and-conquer” approach and recursively divide attributes at each node in the tree based on the information they possess. Although decision tree algorithms are not always the most competitive learners in terms of accuracy, they are computationally efficient and the results produced can be easily converted to human-readable formats.

Classification rules. Rule learning algorithms are used for learning ”IF-THEN” rules. Like decision trees, rule learning algorithms work on training samples with similar formats. However, since the rule-sets learned are disjoint to each other, usually rule learning algorithms produce far fewer rules than decision trees on the same

SenSys’05, November 2–4, 2005, San Diego, California, USA.
ACM 1-59593-054-X/05/0011.



(a) Data delivery rate. (b) Packet latency. (c) Fairness.

Figure 1: Performance comparisons between MetricMap and MT.

training set, with a comparable accuracy. This makes it preferable in scenarios where classifiers need to be used at runtime.

The features we used for link quality learning are listed in Table 1.

features	description	type
`RSSI`	received signal strength indicator	numeric
`sendBuf`	send buffer size	numeric
`fwdBuf`	forward buffer size	numeric
`depth`	node depth from the base station	numeric
`CLA`	channel load assessment [1]	numeric
`pSend`	forward probability	numeric
`pRecv`	backward probability	numeric

Table 1: Feature Vector

2.2 Validation and Results

Our learning and validation experiment is performed on Weka [5], a workbench containing implementations of a variety of standard machine learning algorithms. We use the J4.8 algorithm provided with Weka for decision tree learning and JRip algorithm for classification rule learning. J4.8 implements an improved version of the C4.5 algorithm and JRip implements RIPPER, a propositional rule learner.

We validated the accuracy of link quality classification using 10-fold cross validation. Cross validation is a standard method to estimate classification accuracy over unseen data. The available data is divided into ten equal-sized blocks. Nine of the blocks are randomly chosen and used for training a classifier, with the remaining block used for validation. This process is repeated 10 times in our experiment to give a reliable measure of classification accuracy, which is 82% using J4.8 and 80% using JRip for our evaluation on MoteLab.

3. IMPACT ON ROUTING

In this section, we introduce a routing protocol (MetricMap) using our learning-based link quality assessment method and present results of a MetricMap prototype deployed on MoteLab, a 30-node (MicaZ) sensor network testbed at Harvard University. Due to space constraints, we will not discuss the implementation details of MetricMap. The key idea is to replace the current traffic-based assessment method in MT with our offline collected rule-sets, whenever the former fails to function.

Our experiment consists of two phases: an offline learning phase that takes multiple hours to collect training samples and perform training tasks, and an online optimization phase that uses the rule-sets learned from the offline phase to estimate link quality at runtime. As pointed out in Section 2, we use a classifier produced by JRip algorithm due to its small number of rule-sets. Each experiment runs 15 minutes, with a data packet size of 12 bytes. In our evaluation, we focus on the following three performance metrics: data delivery rate, packet latency and fairness. We use the same fairness definition as in [2].

Figure 1(a) compares the data delivery rate between MetricMap and MT. It shows that MetricMap consistently outperforms MT when the offered traffic loads are higher than 0.5pps. The higher the traffic load, the better MetricMap performs compared to MT. This result is consistent with our observation that MT rarely will find a routing tree under high traffic rates, while MetricMap can because it does not rely on data traffic for link quality assessment. Figure 1(b) shows the packet latency comparison. MetricMap usually has a slightly lower latency than MT, as link quality derived from offline learning does not impose delays on packet transmission. Figure 1(c) compares the node fairness in delivering data packets. It demonstrates that MetricMap does not treat certain nodes better than others. This is reasonable since all nodes use similar rule-sets learned offline and there is no bias towards any particular link. On the other hand, since MT relies on data traffic to infer link quality, the link selected may be skewed depending on the traffic pattern.

4. CONCLUSION

This paper presents a supervised learning based link quality estimation technique and its application on maintaining high performance of metric-based routing. From our preliminary experiment, this method shows promise for high traffic rate applications, with performance improvement in terms of data delivery rate up to 400%, when evaluated on MoteLab. In the future, we plan to apply our approach to other sensor network testbeds with different scales, topologies and hardwares. Other potential areas of future work include using ensemble methods to improve learning accuracy, and extending MetricMap to support dynamic updates of classifiers.

5. REFERENCES

[1] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Procs. of MobiCom*, 2003.

[2] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *Procs. of SenSys*, 2004.

[3] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[4] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell. CODA: congestion detection and avoidance in sensor networks. In *Procs. of SenSys*, 2003.

[5] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition*. Morgan Kaufmann, 2005.

[6] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Procs. of SenSys*, 2003.

[1] Channel load assessment is similar to the one used in CODA [4] to detect local network congestion.