The Impact of Network Topology on Collection Performance

Daniele Puccinelli*[1], Omprakash Gnawali*[2],
SunHee Yoon[2], Silvia Santini[3], Ugo Colesanti[4],
Silvia Giordano[1], and Leonidas Guibas[2]

[1] University of Applied Sciences of Southern Switzerland
[2] Stanford University
[3] ETH Zurich
[4] Sapienza University of Rome

Abstract. The network topology has a significant impact on the performance of collection protocols in wireless sensor networks. In this paper, we introduce an unobtrusive methodology to quantify the impact of the topology on the performance of collection protocols. Specifically, we propose a protocol-independent metric, the Expected Network Delivery, that quantifies the delivery performance that a collection protocol can be expected to achieve given the network topology. Experimental evidence obtained with two collection protocols on numerous topologies on testbeds shows that our approach enables a systematic evaluation of protocol performance.

1 Introduction

The rich and active research in network protocols in Wireless Sensor Networks (WSNs) has progressively emphasized the testbed evaluation of protocols over simulation. Several testbeds exist with a hundred or more mote-class nodes. The use of these testbeds has led to protocols that can function in the harsh environment they often encounter in real-world deployments. The Collection Tree Protocol (CTP) [1], for example, was adopted in several deployments [2][3] due to the promising results it achieved on a large number of testbeds.

Experiments on a testbed subject a network protocol to the vagaries of real-world wireless links [4], with no approximations or simplifying assumptions about their behavior. This is a clear improvement over simulation. The uncertainty in the behavior of wireless links is valuable for protocol evaluation, but it also represents a drawback of testbed experiments compared to simulations. Network simulations offer a fine-grained control of the network environment and the propagation conditions. In contrast, testbed users can only test their protocols on specific, non-reproducible situations, because they have almost no control over the state of the network.

* Lead Authors

In a wireless network, the topology is jointly determined by the *network layout* and the *link dynamics*. The effective topology over which routing paths are established also depends on the choice of routing destination, which corresponds to the *sink placement* in the context of WSNs. The combination of the network layout, the link dynamics, and the sink placement, which we simply refer to as *network topology*, has a large impact on protocol performance.

The performance of a protocol is a function of the topology as well as of the protocol's own mechanisms. Thus, we cannot attribute the performance achieved by a protocol entirely to its mechanisms without considering the state of the network. This makes it challenging to reason about protocol performance on a testbed. Figure 1 shows that both CTP and the Arbutus collection protocol [5] achieve a wide range of delivery ratio and goodput levels even on a single testbed. With both protocols, we observe a dichotomy between high-performing and low-performing topologies, which we refer to, respectively, as Class A and Class B topologies. Due to the lack of a methodology to describe the topology on which a testbed experiment is performed, even in papers where protocols are compared experimentally on real-world testbeds, there is at most a quick comment on the topology used. Different experiments may have been run over different network topologies, which makes it difficult for the community to reproduce the testbed results or to systematically reason about the differences in protocol performance across testbeds.

To cope with the lack of control over the state of a testbed across multiple experiments, we propose to explicitly capture the state of the network while evaluating protocols on the testbeds. For this purpose, we introduce a protocol-independent network metric, the Expected Network Delivery (END), that captures the reliability of the achievable routing paths from each node to the sink. The END quantifies the delivery performance that a collection protocol can be expected to achieve given the network topology. This metric helps decouple the impact of the network topology from the impact of the protocol's own mechanisms on collection routing performance.

Using the END to characterize the network enables a systematic testbed evaluation of network protocols despite the lack of control over the testbed topology. A collection protocol, for example, might achieve different delivery ratios when tested on different testbeds or even on the same testbed at different times. If the END changes significantly across multiple experiments, changes in the network topology can explain the performance variations. On the other hand, if the END remains stable, the difference in performance can be attributed to the mechanisms in the protocol that reacted differently on different experiments despite the network state being roughly the same. Moreover, the range of END values across different experiments captures the range of network conditions encountered during the experiments. If we test a protocol across a large number of testbeds but only span a narrow END range, then we have failed to test the protocol across a wide range of network conditions.

We have run a large number of experiments with two different collection protocols, CTP [1] and Arbutus [5], on the Motelab [6] testbed over a period

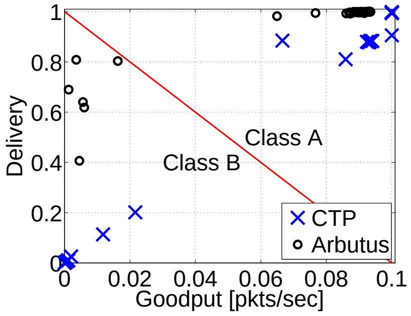

Fig. 1. The performance of collection protocols in MoteLab varies significantly depending on the sink placement, as shown by these results obtained with CTP and Arbutus. There are two distinct performance classes, which we label as A and B.

of several months. Furthermore, we have tested the performance of CTP on the Castalia wireless sensor network simulator [7]. We observed that different combinations of protocol, sink placement, testbed, and experiment time results in a wide range of performance. With the END computed during these experiments, we were able to conclude that the performance variations were primarily due to the properties of the topology present during those experiments rather than the protocol mechanisms.

In this paper, we make these contributions:

- We show that the performance of collection protocols on testbeds depends on the network topology at the experiment time.
- We design the END, a protocol-independent metric to capture the key properties of the network topology that affect the performance of the protocol.
- We propose a methodology to systematically evaluate the performance of a protocol across various testbeds, topologies, and experiments despite having no control over the network dynamics on the testbeds.
- We evaluate the effectiveness of the END, by analyzing the results from a large number of testbed experiments as well as simulations, with CTP and Arbutus collection protocols as examples.
- We show that our methodology is applicable not only to collection, but also to other categories of protocols.

2 Quantifying the Impact of the Topology

In this section we define the Expected Network Delivery, our primary topology-aware metric, along with a secondary metric called Balanced Delivery. We illustrate that these metrics quantify the impact of the network topology on collection performance by capturing the impact of the key links in the network given the node layout and the sink placement.

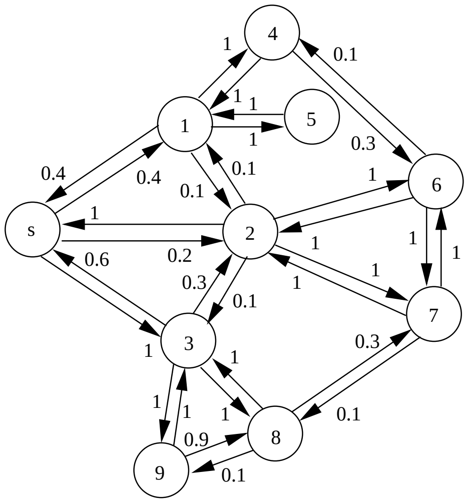

Fig. 2. A sample network with challenging connectivity conditions. The numeric values next to the arrows represent the PRR of the corresponding links in the direction of the arrow.

Node k	**Route**	e_k		
		$\lambda_{i,j} \triangleq \min(\pi_{i,j}, \pi_{j,i})$	$\lambda_{i,j} \triangleq \pi_{i,j}$	$\lambda_{i,j} \triangleq \pi_{j,i}$
1	[1, s]	0.4	0.4	0.4
2	[2, s]	0.2	1	0.2
3	[3, s]	0.6	0.6	1
4	[4, 1, s]	0.4	0.4	0.4
5	[5, 1, s]	0.4	0.4	0.4
6	[6, 2, s]	0.2	1	0.2
7	[7, 2, s]	0.2	1	0.2
8	[8, 3, s]	0.6	0.6	1
9	[9, 3, s]	0.6	0.6	1
		END=0.4	I-END=0.67	O-END=0.57

Table 1. Expected path delivery (EPD) values for the nodes in the network in Figure 2. The EPD is computed in three different ways: with $\lambda_{i,j} \triangleq \min(\pi_{i,j}, \pi_{j,i})$ to obtain the END, with $\lambda_{i,j} \triangleq \pi_{i,j}$ to obtain the I-END, and with $\lambda_{i,j} \triangleq \pi_{j,i}$ to obtain the O-END. The END, I-END, and O-END are computed as the average of the corresponding EPDs.

2.1 Expected Network Delivery

Let $\mathcal{N} \subset \mathbb{N}$ denote the set of nodes in a WSN. We assume a many-to-one traffic flow to a sink $s \in \mathcal{N}$ enforced by an arbitrary distributed routing protocol. When node i transmits to node j, they form a directional wireless link that we denote as (i, j). We use a comma-separated list of nodes within square brackets to denote a route; for instance, if i uses j as a relay to get its packets to s, the corresponding two-hop route is represented as $[i, j, s]$. We define the (one hop) link Packet Reception Ratio (PRR) over the link (i, j), $\pi_{i,j}$, as the fraction of the packets transmitted by i that were directly received by j (*i.e.*, over one hop) over a given time window T. The link PRR values collectively give us a snapshot of the network connectivity over T. We account for asymmetric links by using $\lambda_{i,j} \triangleq \min(\pi_{i,j}, \pi_{j,i})$ as the PRR for the link (i, j). We refrain from using an ETX-like metric such as the product $\pi_{i,j}\pi_{j,i}$ because the forward and the reverse channel are not independent [8].

Given the specific sink placement, each node employs a distributed routing protocol to find a route to the sink. We assume that the protocol's goal is to maximize the delivery of data packets to the sink. To capture the state of the network, we acquire network connectivity data while the protocol is running and, after the completion of the experiment, compute the link PRRs and apply Dijkstra's algorithm [9] with $1/\lambda_{i,j}$ as the link metric to obtain the paths from each node to the sink that maximize the overall delivery to the sink. We then compute the Expected Path Delivery (EPD) e_k from node k to the sink s as

$$e_k = \mathit{\Pi}_{h=0}^{H-1} \lambda_{r_h, r_{h+1}}, \tag{1}$$

where r_h represents the h^{th} hop between k and s (with $r_0 \triangleq k$ and $r_H \triangleq s$), and H denotes the number of links that form the route between k and s. In order to quantify the expected performance of a collection protocol with a global knowledge of the network topology, we define our topology-aware collection metric, the **Expected Network Delivery** (**END**), denoted as $\text{END} \in [0, 1]$, as the expected path delivery averaged over all nodes:

$$\text{END} = \frac{1}{|N|} \sum_{k \in \mathcal{N}} e_k. \tag{2}$$

The END is therefore a function of the link PRRs, which capture the net effect of all the vagaries of wireless propagation. For this reason, our metric captures the ground truth of the state of the network and describes the network topology in a protocol-independent fashion. The END captures the impact of the link connectivity on a network-wide level, distinguishing the key links from the redundant ones. Though in this paper we focus on many-to-one traffic, our methodology is based on the connectivity properties of the network and could be applied to any traffic pattern.

Although our metric is protocol-independent, it is necessary to extract it while a given protocol is running so that we can capture the properties of transitional links during the experiment. If a topology is dominated by unstable links whose coherence time is lower than or comparable to the duration of the experiment, then even capturing connectivity data right before and right after the experiment would be misleading. Since WSN routing protocols typically employ broadcast control traffic for topology discovery and route maintenance,we obtain our metrics by computing the PRR measurements based on the protocol's control traffic, which is acquired over the testbed's backchannel. This approach is non-intrusive, because it relies on passive measurements that do not interfere with the protocol. We ignore all protocol-specific information, such as, for instance, the contents of the neighbor tables.

Since the END is computed by using $\lambda_{i,j} \triangleq \min(\pi_{i,j}, \pi_{j,i})$ as the PRR for the link (i, j), the END is insensitive to the direction of the network traffic. For this reason, we complement the END with a secondary metric, the **Balanced Delivery (BD)**. The BD, denoted as $B_s \in [-1, 1]$, is defined as:

$$B_s = E_s^{(\mathrm{out})} - E_s^{(\mathrm{in})}, \tag{3}$$

where $E_s^{(\mathrm{out})} \in [0, 1]$ is the *Outbound* Expected Network Delivery (O-END) of the sink s, and $E_s^{(\mathrm{in})} \in [0, 1]$ is the *Inbound* Expected Network Delivery (I-END) of the sink s. The I-END is obtained by applying Dijkstra's algorithm with $\lambda_{i,j} \triangleq \pi_{i,j}$, while the O-END is obtained by applying Dijkstra's algorithm with $\lambda_{i,j} \triangleq \pi_{j,i}$.

2.2 Capturing the Impact of the Key Links

We use the network shown in Figure 2 as an example to explain how the proposed metrics are computed. In the figure, the value on each directional link indicates the PRR. In Table 1, we report the optimal route from each node k to the sink s obtained with Dijkstra's algorithm, along with the corresponding expected path delivery e_k with respect to the appropriate link metric. The END, I-END, and O-END are obtained by averaging out the expected path deliveries over all nodes.

The distribution of the link PRR for all the links in the network might seem like a promising alternative to the END. A network with a large number of high quality links should result in a better protocol performance. However, the protocol performance depends on the quality of the links that the protocol uses and not on the quality of the remaining links. We use the expression *key links* to indicate those links whose absence would partition the network or force the routing protocol to use unreliable links. Because efficient and reliable routing protocols select key links, the END is designed to capture their impact. To clarify this point, let us perturb the network in Figure 2 in different ways to see how the END responds as opposed to the mean link PRR.

1. *An unreliable key link becomes reliable.* The improvement of a key link is a huge benefit to the network, and so the value of a valid topology-aware

Experiment set	Testbed	Routing	IPI [sec]	IBI [min]	Points	Duration [hrs]			
						ave	min	max	tot
motelab-ctp	MoteLab	CTP	10	Trickle	18	1	1	1	18
motelab-arbutus	MoteLab	Arbutus	10	1	32	0.5	0.2	1	16

Table 2. Overview of the experiment sets used in the paper.

metric should increase significantly. Link (s, 2) is a key link with a low PRR. If the PRR of this link increases to 1, the END increases by more than 66% (from 0.4 to 0.67), while the mean link PRR only changes slightly (from 0.67 to 0.69).

2. *A reliable key link becomes unreliable.* Link (3, s) is the most reliable key link in the network, although its PRR is just 0.6. If we set $\pi_{3,s} = 0$, the END drops 50% (from 0.4 to 0.2), while the mean link PRR remains virtually unchanged. The drop in the END in response to the worsening of a key link is proportional to the relative importance of the link. For instance, link (3, 8) is only used for route [8, 3, s] and is not as critical as (3, s); if we set $\pi_{3,8} = 0$, the END only decreases by 15% (from 0.4 to 0.34).

These examples illustrate that the added value of the END comes from its ability to distinguish the links that matter from those that do not.

3 Network Topology and Protocol Performance

In this section, we show how the END and BD metrics make it possible to isolate and better understand the impact of the network topology on protocol performance. We also explore the generality of these metrics by applying them to simulation studies of collection and point-to-point routing protocols.

3.1 Experiments and Metrics

Table 2 shows an overview of the experiment sets used in this paper (the results in Figure 1 are from the experiments in motelab-arbutus and motelab-ctp). We also performed a smaller number of experiments on the Tutornet testbed, as well as simulations on TOSSIM [10] and Castalia [7].

In our experiments, each node injects packets at a constant Inter-Packet Interval (IPI) value towards the single destination, the sink. The IPI includes a small jitter to avoid packet synchronization across the nodes. The routing protocols broadcast their own control messages, known as beacons, at a given Inter-Beacon Interval (IBI), which is fixed for Arbutus and variable for CTP, which employs adaptive beaconing [1]. In this study, we use the performance metrics typically employed in evaluation of routing protocols:

Sink	Class	END	Delivery Ratio
46	A	0.73	0.9984
25	A	0.38	0.9864
22	B	0.18	0.8722
90	B	0.05	0.5119

Table 3. END and delivery ratio for four examples of MoteLab topologies.

END Range	Delivery Ratio		Goodput [pkts/sec]	Path Length		Cost	
	mean	σ	mean	mean	σ	mean	σ
[0, 0.5) (B)	0.06	0.09	0.61e-3	7.1	2.5	23.7	18.9
[0.5, 0.9) (A-)	0.9246	1.47e-2	9.2e-2	4.1	0.9	5.5	2.8
[0.9, 1] (A+)	0.9997	2.33e-4	9.9e-2	3.2	0.2	3.7	0.2
[0.5, 1] (A)	0.9361	0.31e-2	9.3e-2	4.0	0.9	5.2	0.9

Table 4. Results from the experiment set motelab-ctp. Performance of CTP at IPI=10s averaged over different END ranges: Class B, Class A, and its subclasses, A+ (upper Class A) and A- (lower Class A).

- Delivery Ratio: The ratio of the number of packets that are delivered to the sink to the total number of injected packets.
- Goodput: Number of application packets delivered to the sink per node per unit time (here measured in pkts/sec).
- Delay: The time it takes for a packet to travel from the source to its destination.
- Cost: The total number of transmissions (including retransmissions) needed to get a packet from its source to the sink.

During each experiment, we log all control beacons and use them offline to measure the PRR for all the network links. In turn, the measured PRR is employed to compute the END metric for the experiment at hand.

3.2 Protocol Performance and Topology

Table 3 shows the values of the END metric and the delivery ratio from four examples of MoteLab topologies from motelab-arbutus that yield very different performance levels. The delivery ratios range from 99.84% with sink 46 to about 51% with sink 90. Across the experiments, there is a distinct correlation between higher delivery ratio and higher END.

END Range	Delivery Ratio		Goodput [pkt/sec]	Path Length		Cost		Delay [sec]	
	mean	σ	mean	mean	σ	mean	σ	mean	σ
[0, 0.5) (B)	0.66	0.15	0.6e-2	5.0	1.5	10.9	3.4	241.7	240.4
[0.5, 0.9) (A-)	0.9967	4e-3	8.8e-2	2.9	0.5	8.9	5.5	1.3	1.5
[0.9, 1] (A+)	0.9996	2e-4	9.3e-2	2.9	0.5	3.3	0.6	0.2	0.1
[0.5, 1] (A)	0.9975	3.6e-3	8.9e-2	2.9	0.5	7.3	5.3	1.0	1.3

Table 5. Results from the experiment set motelab-arbutus. Performance of Arbutus at IPI=10s averaged over different END ranges.

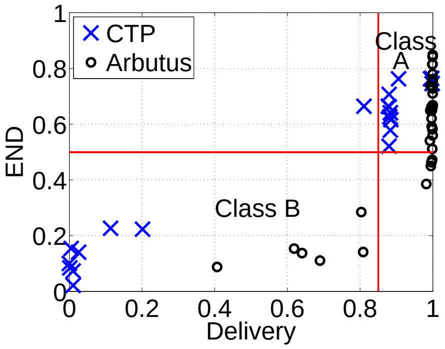

Fig. 3. With both CTP and Arbutus, the END metric generally correlates with the delivery rate, and the performance dichotomy shown in Figure 1 is confirmed. Low END values correspond to a less predictable performance.

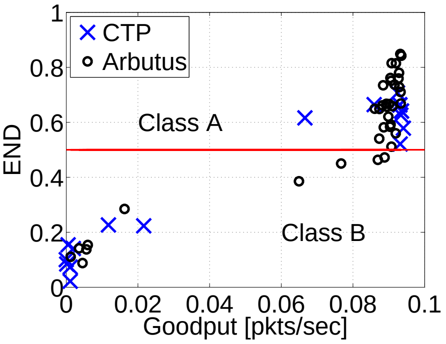

Fig. 4. For both CTP and Arbutus, the END predicts the performance dichotomy between Class A and Class B also in goodput. In general, the better the END, the higher the goodput.

Figure 3 shows the delivery ratio vs. the END metric for the experiments in motelab-ctp and motelab-arbutus. Qualitatively, we observe a correlation between the END metric and the delivery ratio for both CTP and Arbutus; this confirms that the performance variations across different experiments may be traced back to changes in the network topology. With a lower END, the best possible achievable performance is also lower, as reflected in the protocol performance.

Table 4 and Table 5 summarize the results from the experiments in motelab-ctp and motelab-arbutus. We note that the dichotomy between Class A and Class B that was evident in Figure 1 is also visible in these results. We can use the END metric to classify the topologies according to the expected achievable performance. The END also correlates to various degrees with other metrics such as goodput, cost, and delay.

The END metric helps us understand the protocol performance in the context of the network topology over which an experiment is run. With the END metric, we can precisely identify the cases where a low protocol performance is due to an adverse network topology.

3.3 Explaining Protocol Performance

The END metric can help explain the reasons behind the achieved protocol performance. For example, in Class B topologies Arbutus performs more efficiently than CTP due to the use of different retransmission strategies: Arbutus employs unconstrained retransmissions (compared to 32 times for CTP) and limits packet loss at the price of delay, as shown in Table 5. In Class A topologies, CTP and Arbutus perform more similarly and achieve high delivery ratio. We expect this result considering the abundance of high quality key links in Class A topologies.

Figure 5 shows the packet loss and the corresponding END with a specific sink placement (node 22) in MoteLab. Each datapoint represents the packet loss and the value of the END metric over one of 50 experiments. There is a clear negative correlation between the END and the packet loss.

In Class B topologies, a high delivery ratio comes with a high cost because a large number of retransmissions is needed to deliver packets over unstable key links. In Class A, however, a high delivery ratio does not imply a high cost because the key links in Class A topologies have a high PRR and generally require a single transmission.

3.4 Comparisons across Testbeds

The END metric enables a direct comparison of results obtained from different testbeds thereby overcoming the biggest shortcomings in testbed experimentation – the inability to directly compare the results from different testbeds. The results obtained on different testbeds are directly comparable if the END metrics across those experiments are similar.

In one experiment run on the Tutornet testbed, CTP achieved a delivery of 99.9% with an END of 0.89. Both CTP and Arbutus performed similarly in the MoteLab runs from motelab-ctp and motelab-arbutus when the END was in that same ballpark. Because the END values across these experiments on two different testbeds are similar, we know that the network topologies during these experiments were similar, and these two performance results are directly comparable. Thus, the END metric tells us when the topologies on two testbeds are similar and gives us a way to directly compare the results from two testbeds.

3.5 Comparisons over Time

Even if the sink placement and the network layout are fixed, protocol performance can still change over time due to the temporal changes in the link qualities in the network. Figure 5 shows that a testbed can have a time-varying topology that yields a time-varying performance. The performance peaks correspond to END maxima, while the performance lows map to END minima. We conjecture that this is due to the high impact of transitional links with this particular sink assignment: transitional links are more likely to get stuck in bad fading spots at night than they are during the day, when they can leverage induced fading effects [11].

This example shows that the END can also be employed for a systematic evaluation of a single protocol over time.

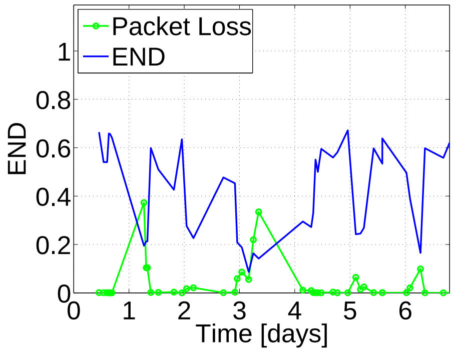

Fig. 5. An unstable sink assignment (MoteLab's 22) results in different network topologies that yield significant performance variations.

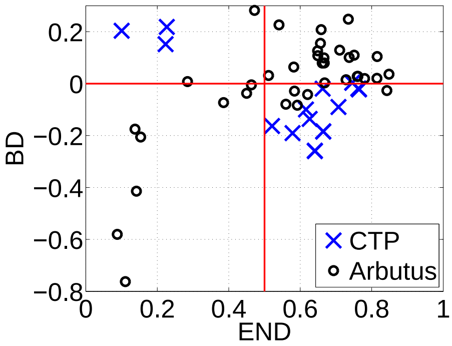

Fig. 6. Experiment sets motelab-ctp and motelab-arbutus: Balanced Delivery vs. END.

3.6 Directionality and Outliers

The quality of the links to the sink's neighbors, to a large extent, determines the performance of a collection protocol. A link can have bidirectional loss, dominantly outbound loss, and dominantly inbound loss. The nodes cannot send data to the sink if the links from these neighbors of the sink have high bidirectional or inbound losses, while acknowledgments and control packets from the sink tend to get dropped with high outbound loss.

Figure 6, shows the BD vs. the END for each of the motelab-ctp and motelab-arbutus experiments. We observe that the outbound losses are common in Class B topologies. Both protocols suffer significant outbound losses in those topologies. Because it contains several mechanisms to boost reliability, Arbutus performs significantly better than CTP with Class B topologies. Figure 6 shows that a near-zero BD always corresponds to a high END and therefore to high delivery ratios. We also found that near-zero BD are rare, suggesting that most links were unstable and asymmetric during our experiments. Strong dominantly inbound losses were never observed in our experiments. The corresponding topologies would result in near-zero packet delivery.

Moderate inbound loss (BD$>$ 0) typically indicates the presence of connectivity outliers. If the END is very high (typically > 0.8), a positive BD is indicative of the presence of leaf connectivity outliers, *i.e.*, nodes with poor downstream links that are attached to the collection tree as leaf nodes. These leaf outliers result in a positive BD in Figure 6. The positive BD allows us to determine that the dominant cause of CTP's poor performance is downstream loss. Thus, the BD metric enhances the performance analysis by adding directionality to the overall topology information captured by END.

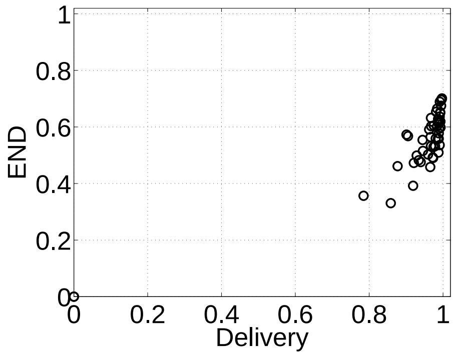

Fig. 7. END vs. delivery in a series of Castalia simulation runs of CTP. Each circle represents the average over 50 simulation runs with a given 100-node network.

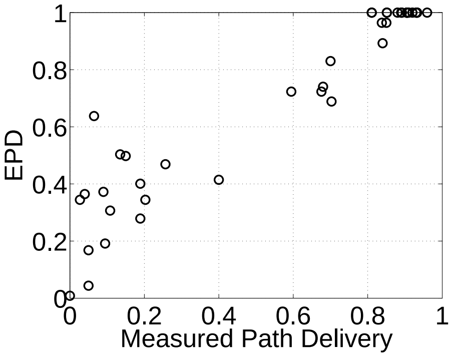

Fig. 8. Expected Path Delivery vs. measured path delivery in a series of TOSSIM simulation runs of TYMO, a point-to-point routing protocol (each circle represents one simulation run).

3.7 Applicability to Simulation

Even an accurate qualitative description of a simulation setup makes it difficult to quickly determine the impact of the simulation environment on the protocol performance. Instead, the END metric can be used to succinctly capture the topology information used in simulations. Figure 7 shows the values of the END and the delivery ratio obtained by running CTP on 50 different network topologies (each consisting of 100 nodes) within the Castalia simulation environment [12][7]. For each network, the END and the delivery ratio were averaged over 50 simulation runs. Similarly to the testbed experiments, the END metric and the delivery ratio from the simulations show a significant degree of correlation. In this specific case, low END values correspond to relatively high average delivery ratios because in the simulations the channel remains constant across retransmissions. Because the END metric succinctly captures the property of the topology instantiated during the simulation, it allows us to understand the impact of the topology on the protocol performance in simulation.

3.8 Applicability Beyond Collection

The definition of the END given in equation (2) presupposes a many-to-one traffic pattern. This formulation is specific to collection, but the framework is more generally applicable. For example, in the case of point-to-point routing, the Expected Path Delivery (EPD) given in equation (1) can be employed to gauge the expected performance on a route between two nodes. Figure 8 shows the results of a TOSSIM simulation of TYMO, a TinyOS implementation of the Dynamic MANET On-demand (DYMO) routing protocol [13]. Each circle represents one simulation run, and each run has a different set of link dynamics. These simulation results show that the EPD correlates well with the measured path delivery, and two performance classes can be identified as was the case with the testbed experiments in Figure 3.

3.9 Limitations

Though the END and the BD are protocol-independent in their definition, their calculation leverages the protocol's control traffic. There is arguably some residual dependence on the protocol, mainly because we need to leverage the protocol's traffic to measure connectivity. Measuring the accuracy of the computed PRR would require the injection of additional traffic, which would affect the protocol's performance and perturb the results. This is a fundamental limitation of our framework that is due to the need to measure the network as we use it [14]. Another limitation lies in the fact that the END is averaged over the duration of each experiment. In networks where bimodal links dominate, or in long experiments, a time-dependent formulation of the END is in order and will be addressed in our future work.

4 Related Work

The impact of the network topology on protocols has been studied in the context of wired networks, with a specific focus on the Internet. Early studies considered the node degree distribution and the neighborhood size [15]. In [16], network topology is characterized with three metrics: the *expansion* (average number of nodes within a given hop count), the *resilience* (minimum cut-set size for a balanced bipartition of the network), and the *distortion* (which captures how path lengths are affected by link failures). Our study, however, is specific to wireless sensor networks, whose low-power communication hardware underscores the probabilistic nature of wireless links [4] and makes it impossible to treat them as Boolean objects (as in wired networks).

Recently, the lack of a *wireless lexicon* to describe the complexities of real-world wireless networks has been pointed out, and there have been a few efforts on the definition of link-level parameters that capture the vagaries of the behavior of low-end wireless network. In [17], a measure of link bimodality (the β factor) is defined, and its impact on protocol performance is characterized. In

[18], a measure of inter-link cross-correlation (the κ factor) is proposed. A related effort is the development of the Stanford Wireless Analysis Tool (SWAT) [19], a software tool for the collection of network measurements. In these studies, network measurements are taken by injecting special traffic patterns: broadcast traffic in [17] and a round-robin of packet bursts in [18]). Our effort can be viewed as complementary to these studies, because (1) we focus on a network-wide metric as opposed to a link-level metric, and (2) we perform passive measurements directly from the broadcast control traffic injected by the protocol under test as it is running. Our approach is particularly valuable for unstable topologies that show different behaviors at different times. Initially, we also attempted to measure the network before and/or after running the protocol, and for unstable topologies the Expected Network Delivery often appeared to be uncorrelated from the various performance dimensions.

Similarly to the CTP work [1] and to the aforementioned studies, we capitalize on remote-access testbeds and their backchannels to gain a thorough understanding of the reasons for packet loss. Similarly to the Visibility framework [20], our approach makes it easier to diagnose the causes of failures. Differently from that framework, however, our approach is unobtrusive because it does not require any changes to the protocol under test. Similarly to [14], we note that testbed conditions vary so rapidly that even back-to-back experiments are not guaranteed to share the same conditions, which is why we measure the PRR from a protocol's control traffic while the protocol is running.

While in [20] visibility is pursued from within the protocol, the achievement of visibility through passive inspection by way of a sniffer network is the focus of [21], [22], and [23]. We believe that coupling our method with passive inspection techniques would greatly benefit the overall system visibility that passive inspection strives for.

This work is informed with a deep awareness of the vagaries of wireless propagation [4], in particular the existence of the transitional region of connectivity [24][25] and the temporal properties of wireless links [8].

5 Conclusion

The wide range of protocol performance levels across different topologies suggests that just looking at the performance results with no regard to the topology only gives an incomplete picture of the protocol performance. The END is a significant step towards a systematic methodology for the comparison of experimental results across protocols, time, and testbeds. We showed that the END exposes specific features of the network topology that can significantly affect the network performance.

The effectiveness of our approach in describing the state of the network during an experiment suggests that it is possible to succinctly represent the network topology in the context of the design goals of a protocol. We primarily focused on collection protocols, but also showed that our methodology applies to point-to-point routing. We underscored its added value in the context of testbed experiments, and we showed that our framework is also applicable to the characterization of simulation scenarios.

Acknowledgements

This work was partially supported by the European Commission under the SCAMPI Project (ICT grant agreement 258414), the Stanford Army High Performance Computing Research Center (grant W911NF-07-2-0027), and the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS) (Swiss National Science Foundation grant number 51NF40-130758/1).

References

1. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection Tree Protocol. In *7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, Berkeley, CA, November 2009.
2. M. Bathula, M. Ramezanali, I. Pradhan, N. Patel, and J. Gotschall. A sensor network system for measuring traffic in short-term construction work zones. In *The 5th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS'09)*, Marina Del Rey, CA, June 2009.
3. J. Ko, T. Gao, and A. Terzis. Empirical Study of a Medical Sensor Application in an Urban Emergency Department. In *4th Intl Conference on Body Area Networks (BodyNets 09)*, Los Angeles, CA, April 2009.
4. K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An Empirical Study of Low-Power Wireless. *ACM Transactions on Sensor Networks (To appear)*, 2010.
5. D. Puccinelli and M. Haenggi. Reliable Data Delivery in Large-Scale Low-Power Sensor Networks. *ACM Transactions on Sensor Networks*, Jul. 2010.
6. G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: a Wireless Sensor Network Testbed. In *4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, April 2005.
7. Castalia - A Simulator for WSNs. `http://castalia.npc.nicta.com.au`.
8. A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin. Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing. In *ACM/IEEE Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, April 2005.
9. E. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.
10. P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In *1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, USA, November 2003.
11. D. Puccinelli and M. Haenggi. Spatial Diversity Benefits by Means of Induced Fading. In *Third IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'06)*, Reston, VA, USA, September 2006.
12. U.Colesanti and S. Santini. A Performance Evaluation Of The Collection Tree Protocol Based On Its Implementation For The Castalia Wireless Sensor Networks Simulator. Technical report, ETH Zurich.

13. I. Chakeres, E. Royer, and C. Perkins. Dynamic MANET On-demand Routing Protocol. IETF Internet Draft – work in progress draft-ietf-manet-dymo-00, February 2005.
14. O. Gnawali, L. Guibas, and P. Levis. A Case for Evaluating Sensor Network Protocols Concurrently. In *The Fifth ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WINTECH'10)*, Chicago, IL, USA, September 2010.
15. M. Faloutsos, P. Faloutsos, and C. Faloutsos. What does the Internet look like? Empirical Laws of the Internet Topology. In *SIGCOMM'99*, Cambridge, MA, USA, September 1999.
16. P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design. Technical Report 00-731, University of Southern California, February 2000.
17. K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. The Beta-Factor: Improving Bimodal Wireless Networks. In *6th ACM Conference on Embedded Networked Sensor Systems (SenSys'07)*, Raleigh, NC, November 2008.
18. K. Srinivasan, M. Jain, J. Choi, T. Azim, and E. Kim. The Kappa Factor: Inferring Protocol Performance Using Inter-link Reception Correlation. Technical Report 09-02, Stanford University, 2009.
19. K. Srinivasan, M. Kazandjieva, M. Jain, E. Kim, and P. Levis. SWAT: Know Your Network. In *8th International Conference on Information Processing in Sensor Networks (IPSN'09)*, San Francisco, CA, 2009.
20. M. Wachs, J. Choi, J. Lee, K. Srinivasan, Z. Chen, M. Jain, and P. Levis. Visibility: A New Metric for Protocol Design. In *5th ACM Conference on Embedded Networked Sensor Systems (SenSys'07)*, Sydney, Australia, November 2007.
21. M. Ringwald, K. Roemer, and A. Vialetti. Passive Inspection of Sensor Networks. In *3rd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '07)*, Santa Fe, NM, June 2007.
22. K. Roemer and M. Ringwald. Increasing the Visibility of Sensor Networks with Passive Distributed Assertions. In *Workshop on Real-World Wireless Sensor Networks (REALWSN'08)*, Glasgow, UK, April 2008.
23. K. Roemer and J. Ma. Pda: Passive distributed assertions for sensor networks. In *8th International Conference on Information Processing in Sensor Networks (IPSN'09)*, San Francisco, CA, USA, April 2009. IEEE Computer Society.
24. A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, November 2003.
25. M. Zuniga and B. Krishnamachari. An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links. *ACM Transactions on Sensor Networks*, 3(2):1–30, 2007.