Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive Lighting in Road Tunnels

Matteo Ceriotti[1,2], Michele Corrà[3], Leandro D'Orazio[4], Roberto Doriguzzi[5], Daniele Facchin[2], Ştefan Gună[2], Gian Paolo Jesi[2], Renato Lo Cigno[2], Luca Mottola[2,6], Amy L. Murphy[1], Massimo Pescalli[4], Gian Pietro Picco[2,*], Denis Pregnolato[4], Carloalberto Torghele[2]

[1]Bruno Kessler Foundation, Trento, Italy [2]University of Trento, Italy [3]TRETEC, Trento, Italy
[4]Siemens Italy [5]CreateNet, Trento, Italy [6]Swedish Institute of Computer Science (SICS)
[*]Contact author: gianpietro.picco@unitn.it

ABSTRACT

Existing deployments of wireless sensor networks (WSNs) are often conceived as stand-alone monitoring tools. In this paper, we report instead on a deployment where the WSN is a key component of a closed-loop control system for adaptive lighting in operational road tunnels. WSN nodes along the tunnel walls report light readings to a control station, which closes the loop by setting the intensity of lamps to match a legislated curve. The ability to match dynamically the lighting levels to the actual environmental conditions improves the tunnel safety and reduces its power consumption.

The use of WSNs in a closed-loop system, combined with the real-world, harsh setting of operational road tunnels, induces tighter requirements on the quality and timeliness of sensed data, as well as on the reliability and lifetime of the network. In this work, we test to what extent mainstream WSN technology meets these challenges, using a dedicated design that however relies on well-established techniques. The paper describes the hw/sw architecture we devised by focusing on the WSN component, and analyzes its performance through experiments in a real, operational tunnel.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—Wireless Communication

General Terms

Design, Experimentation, Measurement

1. INTRODUCTION

Wireless sensor networks (WSNs) are regarded as a fundamental element of a vision where the fine-grained *monitoring* provided by distributed sensing enables accurate, automatic *control* of a physical environment. Nevertheless, real-world deployments of WSNs found in the scientific literature appear to be mostly conceived as stand-alone monitoring-only tools. The data they collect is either funneled to a collection station and made available for offline analysis [17], or is used in-network to detect predefined conditions and generate alarms [15]. The ability to "close the loop" and affect the environment is missing from most of the applications reported.

This paper presents a novel application where WSNs are used precisely to enable closed-loop control, for adaptive lighting in road tunnels. This system has been developed in TRITon (Trentino Research & Innovation for Tunnel Monitoring, `triton.disi.unitn.it`), a project carried out by research centers and companies, funded by the local administration in Trento, Italy, with the goal of reducing the management costs of road tunnels and improving their safety. Our WSN-based control system is to be installed in *operational* tunnels on a high-traffic freeway—an ambitious goal given that WSNs have never been used in this context.

Adaptive lighting in road tunnels. In state-of-the-art solutions, tunnel lighting is either pre-set based on current date and time, or set by an open-loop regulator relying on an external sensor. Both solutions disregard the actual lighting conditions inside the tunnel, and may endanger drivers or consume more power than needed.

In the system we describe here, a WSN deployed along tunnel walls measures the light intensity and reports it to a controller, which closes the loop by setting the lamps to match the lighting levels mandated by law. Unlike conventional solutions, our system adapts to fine-grained light variations, both in space and time, and dynamically and optimally maintains the legislated light levels. This enables energy savings at the tunnel extremities, where sunlight enters, but it is also useful inside the tunnel to ensure the target light levels even when lamps burn out or are obscured by dirt. We detail further the adaptive lighting problem in Section 3.

Motivation for WSNs. We are sometimes asked: *"Why should one use a WSN in tunnels, where power and network cables are already available?"* Although power cables are present along with lighting, realizing the shunts necessary to operate the distributed sensor nodes at the right voltage is expensive. Similar considerations hold for network cables, actually found only in medium and long tunnels. Finally, the untethered WSN nodes can be placed *anywhere* along the tunnel—i.e., where lighting engineers say it is best to sense light—and not only where cables already exist.

A WSN solution drastically reduces installation and maintenance costs, especially when the target is an *already-existing* tunnel, where changes to the infrastructure should be minimized. This is often the case in Trentino, the province managed by the administration funding TRITon, a mountainous area of 6.200 km^2, 500,000 people, and over 150 tunnels for a total of 50 km, the majority of which

IPSN'11, April 12–14, 2011, Chicago, Illinois.

are old and under 500 m. In these tunnels, a small investment can significantly improve safety and reduce energy bills.

Challenges. As we discuss in Section 4, tunnels are harsh environments, relatively well-studied but for which real-world WSN experiences are largely missing. In our case matters are complicated by vehicular traffic, which affects wireless communication, and light itself, which is notoriously difficult to measure accurately and yet whose (abrupt) variations are the essence of our application. These challenges notwithstanding, the practical goal of TRITon is to deploy a WSN-based adaptive lighting system in a 630 m, two-lane, double-carriageway operational tunnel with an average traffic of more than 27,000 vehicles per day. The design decisions for the WSN supporting closed-loop control in such a safety-critical environment are dominated by real-world constraints, including:

1. extended lifetime is paramount: changing batteries can be easily performed during tunnel maintenance, but tunnel operators expect *at least* a 1-year lifetime;
2. continuous operation implies that the WSN cannot fail: node failures are important, but sink failures are critical;
3. sensed data must arrive timely: we do not face hard real-time constraints, yet delays induced by node and communication failures may jeopardize control;
4. the quality of sensing impacts directly the quality of control: sensor accuracy and noise reduction are key;
5. integration with conventional, industrial-strength equipment poses complex engineering challenges.

Contribution. We deliberately choose to tackle the challenges above by reusing existing techniques whenever possible, as the target scenario already entails several complex engineering and deployment issues. However, the staple WSN mechanisms and protocols in monitoring-only deployments have essentially never been tested in such a challenging setting, also including closed-loop control. Bearing this in mind, our contribution lies precisely in:

1. verifying that a WSN-based solution to adaptive lighting is *feasible* in road tunnels;
2. understanding to what extent the solution can be achieved by relying on *mainstream* WSN technology;
3. identifying a *combination* of techniques, among the many reported in the literature, successful in our peculiar setting;
4. demonstrating the above in an *operational* testbed where the WSN is integrated with standard tunnel equipment.

We also believe that gaining practical insights into the aspects above reaches beyond the specifics of our road tunnel scenario. Some of the requirements we are forced to cope with are akin to related scenarios where the use of WSNs is envisioned but only partly accomplished; for example, metropolitan subways [2], underground mines [13], and service pipes [25]. The real-world lessons we learned may be an asset for the designers of these systems.

Section 5 illustrates the system architecture by concisely describing each functional component. The focus of the paper is however on the WSN one. Section 6 describes how we tackled the aforementioned challenges by relying on a popular platform: TelosB-like motes running TinyOS. The motes host custom-made sensor boards we calibrated for our tunnel setting. The software deployed on the motes includes dedicated communication protocols, whose design however relies on the combination of well-known techniques. A distinguishing aspect of our software layer is that both application and system-level services (e.g., routing) are built atop middleware [5] that, compared to using directly the operating system as in the vast majority of reported WSN deployments, greatly reduces the programming effort and yields a smaller binary footprint.

The high volume of vehicular traffic in our final tunnel deployment prevents us from using it for experimenting with parameters and performing validation tests. Therefore, in this paper we report on results in a second tunnel that, although operational, is less trafficked and offers a more flexible experimental testbed to analyze and tune our system, which must work right away upon installation in the final tunnel. The equipment we installed is described in Section 7. The testbed experiments, over a 7-month period, are reported in Section 8, where we analyze both the quality of control and the WSN performance. Results show that our system accurately closes the control loop even in the presence of noisy and inappropriate lighting equipment. Moreover, they confirm that the WSN meets the above challenges by guaranteeing a 99.98% data yield, a reporting delay compatible with the operation of the control system, and an (under-)estimated lifetime well beyond a year.

Section 9 concisely reports on experiments hinting at the fact that the WSN we designed for adaptive lighting can be reused effectively to detect fire, with only very minor modifications. Section 10 ends the paper with brief concluding remarks.

2. RELATED WORK

The literature related to this work concerns the use of wireless technology, including WSNs, in tunnels and similar environments, and the design of closed-loop control systems relying on WSNs.

Wireless technology in tunnels. The behavior of wireless transmissions in tunnels and similar environments has been studied extensively, e.g., for what concerns path loss [26] and radio propagation [19]. Existing works show that the shape of tunnels determines an "oversized waveguide" effect [19]. As for WSNs, Mottola et al. [20] compare the wireless topology of two tunnel deployments against a vineyard one. Section 4 summarizes some of their findings, which we considered in our design choices.

Existing WSN applications in road tunnels focus on monitoring for emergency services [27] and disaster management [4]. These, however, are sophisticated proof-of-concept systems, not designed to sustain long-term operation like the one we present here. WSNs have also been applied in tunnel-like environments, including subways [2], coal mines [13], and service pipes [25]. However, none of these systems involves closed-loop control, and integration with existing, industrial-strength infrastructure is usually not an issue. These are instead some of the characterizing features of our work.

WSN-based closed-loop control systems. Few WSN experiences involve closed-loop control. Lynch et al. [16] integrate a WSN with a semi-active damper to mitigate the structural response of civil infrastructures during earthquakes and similar phenomena. Singhvi et al. [24] rely on mobile nodes to acquire information on the users' behavior and context, to perform adaptive lighting in buildings. Both works focus almost exclusively on the design and optimization of the control algorithms. In contrast, the safety concerns and practical deployment issues concerned with an operational setting play a fundamental role in our work. Kim et al. [11] deploy five wireless sensing stations to perform feedback-driven site-specific irrigation. Their setup is much simpler than ours: each sensing station enjoys permanent power, communicates directly with the base station, and is mapped to a single actuator. Han et al. [8] rely on a WSN to drive the operation of a numerical simulator for plume detection and movement prediction. However, unlike our system, the control loop is entirely within the software realm, and does not affect the physical environment. Finally, Park et al. [21] report on a WSN design for closed-loop light control for entertainment and media production. While the goal of their system is somewhat more sophisticated than ours, their implementation is limited to a small-

scale lab proof-of-concept, which therefore is not confronted with the complexity and engineering challenges of a *long-term, operational* system in a *real-world* environment, which is instead one of the defining features of the work reported here.

3. PROBLEM AND APPROACH

Designing an appropriate lighting for roads is challenging, as it directly affects safety and requires huge amounts of energy. Tunnels inherit these challenges and pose additional ones. Illumination varies significantly along a tunnel's length, unlike on roads, and requires a more sophisticated control in response to environmental conditions. Moreover, and most importantly, the light conditions at the entrance must match closely the external ones to ensure that drivers can still discern obstacles when entering the tunnel.

Satisfying this latter requirement during daytime hours has a huge impact on energy consumption. Indeed, daylight is several orders of magnitude larger than that sufficient for night vision, due to the ability of the human eye to adapt to darkness. To get a concrete feel of the values at stake, solar light may reach in excess of 100,000 lx while night road illumination is usually 5-10 lx. Therefore, the initial few meters of a road tunnel can easily consume in daytime hours the equivalent of kilometers of road lighting at night. On a broader perspective, the 150+ road tunnels in Trentino consume 20 GWh per year, as much as 16,000 people in the same region. Therefore, even a small improvement of the tunnel lighting system can return significant savings on the energy bills.

Tunnel lighting must abide by an illumination curve defined by law [3], that specifies the light level as a function of the distance inside the tunnel, as shown in Figure 1. At the entrance, the curve aims at ensuring continuity of light conditions from the outside to the inside, to avoid that drivers perceive the tunnel as too bright or too dark. As the distance from the entrance increases, the light level is allowed to decrease, as the human eye adapts to darkness.

Conventional solutions. The legislated curve is currently met by simple solutions that over-approximate the safe light levels, therefore wasting energy. The most common solution is a simple timer, that automatically sets the light intensity along the tunnel based on date and time, entirely oblivious of the conditions inside or outside the tunnel. More sophisticated solutions employ an open-loop regulator relying on an external sensor: the light setting is based on the outside conditions, but the inside ones are still disregarded.

The desired illumination levels are achieved by relying on two separate circuits. The first one (*permanent lighting*) guarantees a constant illumination and is always on. The second circuit (*reinforcement lighting*) provides the extra light necessary to match the daytime external light, and is therefore normally switched off at night. As lamps are typically set in groups, the curve generated by the reinforcement has a step-wise shape. Each step is typically set well above the target legislated curve—it is not uncommon to see

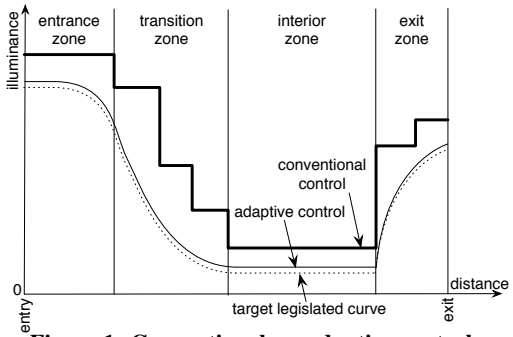


Figure 1: Conventional vs. adaptive control.

designs that over-approximate by a factor of two. Given that pre-set and open-loop solutions lack information from inside the tunnel, this conservative choice ensures a safety margin accommodating the aging of lamps (which reduces their light) and other problems such as burned out lamps. However, it clearly induces a waste of energy, shown in Figure 1 as the area between the step-wise curve used in conventional systems and the one mandated by law.

Closed-loop adaptive control. The figure shows a third line, representing our closed-loop adaptive control where light measurements inside the tunnel are used as feedback to tune the intensity of the lamps, which are *individually* controlled. The knowledge of the actual conditions *inside* the tunnel is the key to *dynamically* match the legislated curve without unnecessary, costly over-provisioning. This knowledge allows us to reconcile the goals of high safety and low energy consumption, unlike pre-set or open-loop solutions. Moreover, knowledge of the inside conditions enables us to leverage natural light to reduce consumption at the entrance—the largest energy drain. Indeed, sun rays entering the tunnel may contribute enough light to push further inside the point at which the artificial lighting becomes necessary. To achieve optimal, dynamic control of the tunnel lighting three "components" are required:

1. An external sensor measuring the *veil luminance*, i.e., the contrast between the tunnel entrance and its background. This parameter is used by regulations to define when a driver can be negatively affected (e.g., dazzled) by the tunnel lighting.
2. A grid of light measurements along the tunnel length, used to compute the error between the legislated target curve (determined as a function of the input veil luminance) and the actual lighting conditions in the tunnel.
3. A control algorithm to drive the above error to zero.

The first component recently became available on the market. The internal measurement system is the main contribution of this paper, reporting on a WSN-based solution. Finally, the design of a control algorithm for adaptive lighting is complicated by the high number of individually-controlled lamps and the mutual influence between these and the sensors. Before presenting our system architecture, however, we describe the characteristics of our scenario.

4. PECULIARITIES OF TUNNELS

Road tunnels are largely unexplored by WSN deployments. Tunnels are harsh environments, where dirt and dust accumulate rapidly and therefore affect sensing, as we discuss in Section 6.2. Periodic tunnel cleaning constitutes an additional threat for the nodes, as it is often performed using high-pressure jets of aggressive detergents. The node packaging is therefore of paramount importance, as it must also meet the general tunnel regulations, e.g., concerning resistance to fire. Vehicular traffic further complicates matters, as the metallic vehicles create interference with the WSN radios, and create occlusions and noise to the light sensors. Moreover, traffic limits access to the tunnel for deployment and debugging purposes, as each visit requires blocking one lane, if not the entire tunnel.

We touch on some of these issues in the rest of the paper. Hereafter, instead, we focus on two aspects that are key to understand our contribution: the characteristics of light in a tunnel environment and how the tunnel shape affects wireless communication.

Light variations. Light is a physical quantity whose precise measurement is already very difficult per se. Tunnels introduce an additional complication, as the light levels vary greatly along its length. Figure 2 shows some of the high-rate (5 s) light measurements we collected for one of the tunnels in TRITon, to understand the light variations and properly design the on-board management

of the sensed data. Distance from the entrance determines how much the external sunlight affects the reading, with clouds and direct sunlight contributing to the largest daily variations. For example, on the second day shown in the figure, direct sunlight entered the tunnel at sunset, causing readings to go off the scale of this chart. Deeper inside the tunnel, the artificial lights have instead the most influence. In Figure 2 one can clearly see the steps caused by changes in the light levels set by the (conventional) control system. Moreover, vehicle headlamps produce transient high readings (barely visible in the night portions of the chart) while trucks occlude sensors and cause the dips visible in the figure. The sensor uncertainty, combined with these phenomena, suggests on-node processing for properly filtering and compensating the data before relaying them to the control system, as we discuss in Section 6.3.

Communication. Previous work by Mottola et al. [20] compared the communication in two tunnels (with and without traffic) against a more conventional outdoor vineyard deployment. Tunnels enjoy better connectivity (i.e., longer links) than outdoor, due to the waveguide effect. This solves and creates problems: better connectivity improves robustness, but also increases the probability of packet collisions. Moreover, according to [20], the network links are more stable in tunnels w.r.t. what is typically reported in the literature for more conventional environments, therefore impacting the relative performance of link estimators. Even in the presence of vehicular traffic, both intermediate and high quality links are accurately identified, and there is a stronger linear correspondence between LQI and packet error rate. As a consequence, LQI performs in tunnels similarly to popular choices such as ETX [6]. We confirmed these findings in the tunnels described here and therefore, as discussed in Section 6, our routing solution relies on LQI.

5. SYSTEM ARCHITECTURE

The functional components of our closed-loop solution for adaptive lighting are shown in Figure 3. The system relies on the WSN for acquiring dense light measurements in the tunnel and wirelessly relaying them in multi-hop to a gateway. Multiple gateways are deployed, to reduce the network diameter and provide redundancy against gateway failures. The gateways forward the sensed data to a Programmable Logic Controller (PLC), the "brain" of the control system. The PLC takes as input the value of the veil luminance measured by an external sensor, along with the data from the WSN. The former is used to determine the target reference lighting, while the latter is used to measure the error from the reference. The PLC directly actuates the lamps to reduce the error and meet the legislated lighting curve. The PLC has access to all the equipment in the tunnel, and can be supervised through the Supervisory Control And Data Acquisition (SCADA) component. PLC, gateways, and SCADA are interconnected by a standard industrial Ethernet LAN,

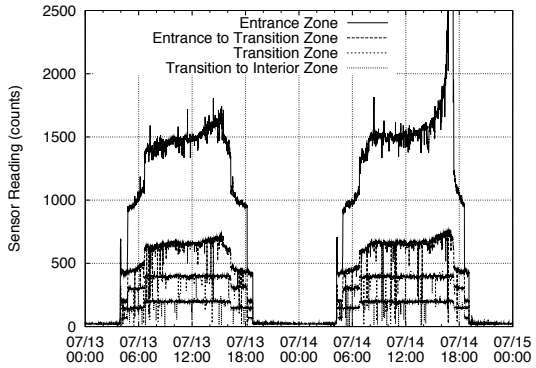


Figure 2: Light levels inside the tunnel over two days.

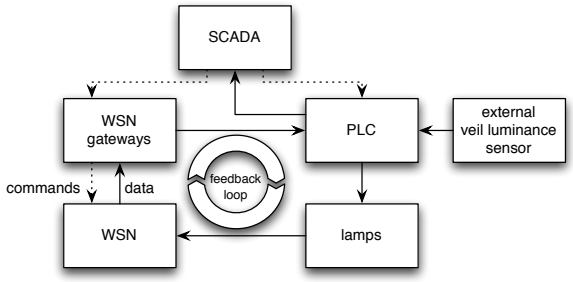


Figure 3: Functional components of the architecture.

running a firewalled TCP/IP suite. The lamps and the veil luminance sensor are instead connected as peripherals of the PLC.

Our system is designed with fault-tolerance in mind. We already mentioned that multiple gateways provide redundancy in the WSN. In the unlikely case that all gateways fail, the PLC switches to an open-loop control relying solely on the external sensor. If this fails too, the PLC defaults to a pre-set lighting curve guaranteeing safety.

As this paper focuses on the WSN component, the corresponding hardware and software architecture is described separately and in more detail in Section 6. The rest of this section illustrates the remaining components to the extent necessary for this paper.

Veil luminance external sensor. The regulations determine the lighting inside the tunnel based on the veil luminance at the entrance. The latter requires a dedicated external sensor, in our case a device by Reverberi Enetec designed specifically to operate in road tunnels. The device is based on a camera-like 1.3 Mpixel CCD sensor, whose output is sent to the device's CPU. The veil luminance value is computed according to regulations [3] and output as an analog 4-20 mA line signal delivered to the PLC, described next.

PLC and control logic. In harsh environments like tunnels, the control functionality is usually implemented by means of a PLC. Its computation is cycle-based: tasks are executed within each cycle, based on the timing requirements of the control algorithm. The hardware and computing power of the PLC depends on the complexity of the tasks and on the number of I/O variables that the PLC must acquire and control. We use a Siemens SIMATIC S7-400H with redundant CPUs, equipped with the appropriate peripherals.

The control logic of the PLC is complicated by the following:

- the number of lamps is large (even hundreds depending on the tunnel) and each must be controlled independently;
- the number of measurement points is also large, to enable a dense-enough sampling of illumination;
- a sensor is affected by many lamps and a lamp affects many sensors, requiring the computation of a complex transfer function from each lamp to each sensor.

This scenario defines a multi-in, multi-out control problem that is highly under-determined, i.e., with fewer measured inputs than controlled variables. Although the control logic is not the focus of this paper, we briefly summarize the problem to the extent allowed by space limitations, to properly place our contribution in context.

Let $\boldsymbol{\Phi} = [\phi_1, \ldots, \phi_L]$ be the vector of light flows from each lamp, $\boldsymbol{M} = [m_1, \ldots, m_M]$ the sensed light measurements, and

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1L} \\ h_{21} & h_{22} & \cdots & h_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M1} & h_{M2} & \cdots & h_{ML} \end{bmatrix}$$

the transfer function of the light intensity from each lamp to each sensor. We can define $\boldsymbol{M} = \boldsymbol{H}\boldsymbol{\Phi}^T + \boldsymbol{N}$ as the set of measured

light samples, where $\boldsymbol{N}$ is a vector of additive noise samples affecting the sensors' measurements, and $\boldsymbol{R} = \boldsymbol{H}\boldsymbol{\Phi}_0^T$ as the target reference working point, computed based on the external sensor and the tunnel lighting standards. The difference between the two, $\boldsymbol{\Delta} = \boldsymbol{M} - \boldsymbol{R}$, represents the error between the target lighting and the one actually measured. The control problem consists of identifying $\boldsymbol{\Phi_0}$ and actuating the lamps to obtain it. Due to the noise term $\boldsymbol{N}$, the direct and exact computation of $\boldsymbol{\Phi_0}$ is not possible, and we resort to minimizing the mean square error

$$\boldsymbol{\Phi_0} = \underset{\boldsymbol{\Phi}}{\operatorname{argmax}}(E[||\boldsymbol{\Delta}||^2])$$

where $||$ is the standard Euclidean norm.

The minimization problem above is a convex hull by construction, since all coefficients in $\boldsymbol{H}$ are non-negative and $\boldsymbol{\Phi}$ is strictly positive. The solution can thus be obtained by employing either a Least Square Error (LSE) or Recursive Square Error (RSE) technique [14]. The complexity of LSE is $O(L)$ in the number L of lamps, and leads to a simpler implementation, potentially allowing us to use a cheaper PLC. On the other hand, RSE is $O(L^3)$ but its convergence is faster thanks to the presence of "memory" in the form of an integral component in the control loop. The problem, however, is further constrained by the fact that the light intensity of lamps cannot be set arbitrarily high, and is bound to a maximum value which depends on the lamp characteristics and technology. We are currently evaluating through in-field experiments in our testbed which approach provides the best trade-off among performance, noise sensitivity, and implementation complexity.

Lamps and actuators. The lighting of our final tunnel includes LED (Light Emitting Diode) and HPS (High Pressure Sodium) lamps. The latter, recognizable by their yellow light, are commonly used in road lighting due to their high emission and relatively low consumption. They can be controlled only within 30% to 100% of their illumination range, and changing their intensity takes minutes. LED technology appeared only recently in tunnel lighting. Its white light enables better vision, the lamps have much lower energy consumption, and can be controlled over their entire illumination range almost instantaneously. Our LED lamps have been developed specifically for TRITon. However, in the testbed deployment described in Section 7 we only had HPS lamps installed.

Lamps are controlled *individually* by the PLC through a digital bus, which enables setting illumination precisely at the level required. Instead, conventional solutions control large sets of lamps at once, yielding constant illumination over long sectors of the tunnel with the consequent energy waste mentioned in Section 3.

SCADA. The overall system is completed by a SCADA subsystem connected to the PLC. The SCADA provides an interface to a human operator e.g., to visualize alarms, manually force light settings, and perform other configuration and management tasks remotely. The SCADA also logs all the data coming from the PLC, for statistical as well as legal reasons. In TRITon, we customized the SCADA to be able to access directly the gateway and therefore the WSN, e.g., to collect data and status from the sensor nodes, as well as change configuration parameters such as the sampling rate.

6. WSN ARCHITECTURE

We illustrate the WSN hardware, the calibration of sensors, and the architecture of application and communication protocols.

6.1 Hardware

WSN node and sensors. We used WSN nodes functionally equivalent to TelosB motes [23], equipped with an MSP430 microcontroller, a Chipcon 2420 radio chip, and an on-board inverted-F microstrip antenna. We did not use an external antenna since the on-board already performs well in tunnels, as mentioned in Section 4.

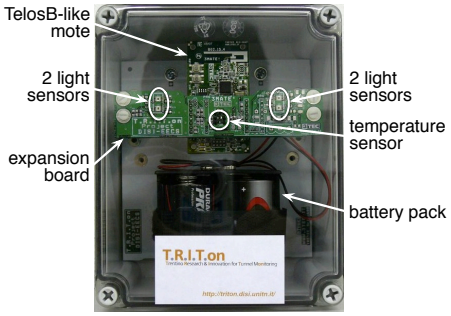


Figure 4: A deployment-ready WSN node.

Our nodes are however more easily expandable than TelosB, thanks to three external connectors that simplify the integration of sensor modules as expansion boards. This proved useful in our case, where we custom-designed an expansion board for the needs of TRITon. The board contains 4 ISL29004 digital light (illuminance) sensors, and 1 TC1047A temperature sensor. As shown in Figure 4, the board is mounted orthogonally to the node, in a "butterfly" configuration. The 4 light sensors are spread in pairs across the horizontal dimension of the board. As discussed in Section 6.3, the reading reported to the gateway is a (filtered) average of the 4 sensor readings. The temperature sensor is not required by light control, but it is useful to analyze the WSN behavior, especially w.r.t. battery discharge phenomena, as we discuss in Section 8.2.

Node and sensor board are powered by 4 Duracell Procell D-size batteries. All the above is packaged in a certified IP65 (water and fireproof) polycarbonate box with a transparent cover.

Gateways. We used a Verdex-Pro embedded computer by Gumstix, equipped with two expansion boards providing Ethernet, USB, and RS232 ports. The former is used to ensure connectivity with PLC and SCADA, while the communication between gateway and WSN sink occurs through the on-board USB port. The RS232 ports are used for debug purposes only. The operating system (Embedded Linux) and applications are stored on the 32 MB flash memory and use the 128 MB of RAM. A 1 GB microSD card provides additional storage. This hosts the database logging WSN light samples, which can be queried directly by the SCADA, and is also useful for debugging the WSN. The database size depends on the number of nodes: for a 20-node WSN it can easily reach 100 MB in a week.

6.2 Calibration of the Light Sensors

We must ensure that the off-the-shelf light sensors are accurate and precise enough in our tunnels. We verified that their response is linear, reducing the calibration task to determining the "right" coefficient α converting from the raw sensor (count) readings to lux. However, the calibration is still complex because *i)* light measurements suffer from a non-negligible, intrinsic uncertainty—at least $\pm 10\%$ in real-world, non-controlled environments—and *ii)* the calibration factor α strongly depends on the light source employed.

Our laboratory setup included professional equipment (e.g., mechanical, high-precision positioners) to guarantee uniformity of the light source (an HPS lamp) and fine control over the light emission. The setup is shown in Figure 5. As a reference gauge we used an ILT1400A radiometer along with its companion SCL110 illuminance probe. Both products are NIST-complaint.

We based the calibration measures on 10 sensors, randomly selected. The measured response included the entire sensor board circuitry, as the integration of the ILS29004 sensor may affect its response. The measurements yielded a value $\alpha = 0.596$, with a

determination factor $R^2 = 0.9986$. We estimated the uncertainty at the end of calibration as $s = \pm\sqrt{(2\sigma)^2 + s_g^2 + s_{NIST}^2}$ where $\sigma = 3.2\%$ is the standard deviation of the uncertainty between the measures from sensor and reference gauge, $s_g = \pm 4,4\%$ is the uncertainty of the reference gauge, and $s_{NIST} = \pm 0,5\%$ is the intrinsic uncertainty of the gauge w.r.t. NIST standards. Our measurements yield a total uncertainty $s = \pm 7,8\%$.

Impact of cover and dirt. The measurements above were carried out by exposing the sensor directly to the light source. However, once deployed, the sensors receive light through the transparent cover of the package described in Section 6.1. Dust and other agents quickly deposit a dirt film over it, especially at the entrances where nodes are easily splashed by water and mud.

Determining how these factors affect light readings is key to ensuring that adaptive control relies on the *actual* tunnel conditions. We performed experiments with clean and dirty covers, comparing the results with the previous ones. We used dirty covers from one of our testbeds on a high-traffic road, after an entire winter with heavy rain and snow. The results are shown in Table 1. A clean, transparent cover is enough to cause a 10% attenuation. The incidence angle does not affect this value significantly. Dirt induces an additional 20-30% attenuation: Table 1 shows two of the dirtiest covers. Although tunnels are periodically cleaned, we are studying ways to prevent or mitigate attenuation, including online calibration and repellent coating.

	Attenuation
Clean cover	9,3%
Clean cover (45°)	10,8%
Dirty cover (A)	30,7%
Dirty cover (B)	33,9%

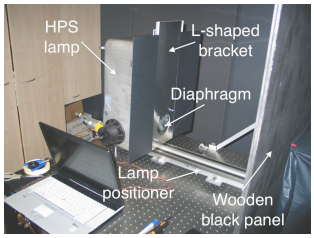
Table 1: Impact of cover and dirt on light readings.

6.3 Software and Communication Protocols

The software and communication protocols of our WSN rely on TinyOS [9]. However, unlike the vast majority of deployments where application and system software sit directly on the operating system, we built the TRITon software on top of the TeenyLIME [5,18] middleware. Our choice was motivated by the fact that TeenyLIME has already been used successfully in another long-term, real-world deployment [1], where its higher-level abstractions have been reported to reduce the overall code footprint, allowing one to pack more functionality on memory-restricted nodes.

Overview. Figure 6 illustrates the architecture of the WSN software. All macro-components in the TRITon application sit atop TeenyLIME, which offers a one-hop shared memory space abstraction in the form of a tuple space, i.e. a collection of typed sequences of fields, on which components can insert, query and receive notifications regarding the data. TeenyLIME both replaces the default message passing communication constructs provided by TinyOS, and bestows upon the architecture a flat layout. Indeed, both application (e.g., sensing) and system components (e.g., data collection) lie on the same level and interact exclusively through the tuple space. We next offer details on these components.

Sensing. The control algorithm requires input values every 30 s from nodes in the entrance, transition, and exit zones, where solar



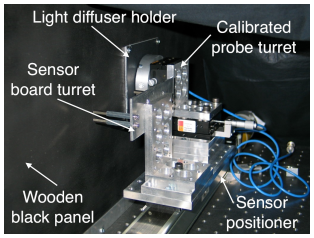


Figure 5: Setup for the calibration of light sensors.

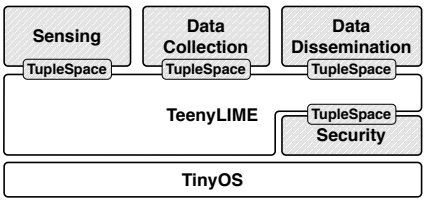


Figure 6: Software architecture.

light has a higher influence, and every 5 min from the nodes in the interior zone. During these intervals, the sensing component collects samples from all 4 light sensors at a configurable rate, set to 5 s by default. As described in Section 4, vehicles produce transient noise—abnormally low or high readings caused by big vehicles and headlights, respectively—that must be filtered out. These outliers are eliminated as follows. Each time a sample is taken, the value of the 4 sensors is averaged into s_i, by excluding the saturating ones, if any. When reporting to the sink, the average s_{all} of all the values s_i is computed. For each s_i, if the difference $|s_{all} - s_i|$ differs from s_{all} by more than 50%, s_i is discarded and s_{all} recomputed.

Data collection. Single-sink collection trees, the most common solution in the literature, are less effective in tunnels. The tunnel linear shape yields a larger network diameter: nodes closer to the root bear a heavier load, funneling information from a large portion of the WSN. Moreover, the larger number of hops increases the probability of data loss between the leaves and the root. Reliability is complicated further by the permanent asymmetries present in tunnel links [20], reducing the effectiveness of the link-layer acknowledgments commonly used to ensure successful transmission.

To provide load balancing and mitigate the risk of message loss on long, multi-hop paths, we adopted a solution collecting data at multiple sinks, ideally spread evenly along the tunnel. Each sink periodically and independently builds a collection tree by flooding a control tuple containing path cost information. The latter is determined by aggregating the per-hop LQI, similar to Multihop-LQI [29], as this technique in tunnels produces overlays similar to those obtained with ETX-based protocols but with much less overhead [20]. Sink selection occurs implicitly, as in CTP [7], by choosing as parent the neighbor with the smallest node-to-sink routing cost. The tree is periodically reconstructed with a sink-initiated message. This allows the routing topology to adjust to topology changes and simultaneously serves as a "keep-alive", enabling nodes to detect when a sink is no longer available. This, together with the implicit sink selection scheme above, provides an automatic hand-over functionality in case a sink fails.

Data reliability is achieved with a hop-by-hop recovery scheme. Data tuples contain a sequence number; upon forwarding, a small number of tuples are cached in the tuple space. When a communication failure occurs, the parent identifies a gap in the sequence and "pulls" the missing tuple from the child's cache.

Dissemination. The WSN nodes can be configured remotely by exploiting one-to-many communication from the gateways, e.g., to change the light sampling frequency or modify MAC parameters. This is useful both to implement the functionality necessary to the SCADA and to manage experiments on the WSN. To disseminate the necessary configuration commands from the gateways to the WSN nodes we employ a Trickle-like scheme [12], which guarantees eventual consistency of the information available at all nodes.

Security. In our environment, *physical* security is the greatest concern, since the WSN nodes are easily accessible and can be easily damaged or stolen. Nevertheless, we designed a simple message authentication scheme based on dynamically-distributed symmetric keys, to ensure that light readings come from legitimate nodes. This component, which at the time of writing is still under test and

integration, is placed between the operating system and the middleware, effectively providing a secure channel on top of TinyOS.

7. TESTBED DEPLOYMENT

As mentioned in the introduction, our final deployment is on a high-traffic road. Carrying out experiments in this site would be impractical, due to the need to block the road partially or totally, and also risky, given that we affect illumination, a key constituent of road safety. Therefore, we were granted access to a shorter, lower-traffic tunnel that served as a testbed we could more easily access to setup our experiments. However, the downside is that we were allowed to replace only partially the tunnel lighting infrastructure. The tunnel is a 260 m-long, two-way, two-lane tunnel. Neither automation nor communication infrastructure was present.

Figure 7 illustrates the equipment we deployed to match the architecture described in Section 5, along with the positions of the various devices. We replaced the first 16 HPS lamps in one of the lanes with 9 250 W HPS lamps (used as reinforcement) and 7 100 W HPS lamps (used as permanent), shown in the figure as dark and white rectangles, respectively. Each lamp is equipped with a ballast containing the electronics necessary to control the emitted luminous flux. These are the only lamps we can control: the others, shown as dashed rectangles in Figure 7, are set by the pre-existing infrastructure through a simple timer.

The lamps are controlled by the PLC, housed in an industrial rack at the tunnel entrance. The PLC bases its decisions on the data collected from the WSN, which contains 40 nodes. The nodes are split evenly between the tunnel walls, and placed at a height of 1.70 m, compatible with regulations. It is important to note that the spacing among nodes, shown in Figure 7, is driven more by the need to stress-test the WSN and the rest of the system rather than to *optimally* close the control loop. Indeed, in our final deployment the position of the WSN nodes is determined by lighting engineering considerations. These are difficult to derive in our testbed because we manage only a fraction of the lighting system in the tunnel. Therefore, the number of WSN nodes is higher and their placement denser than needed. The PLC relies only on the first 15 nodes: the others are extra, used for our experiments. On the other hand, we are putting ourselves in a situation that is *worse* than the one we will find in the final deployment. Indeed, while in the latter we plan to have 44 nodes over 630 m in each pipe, in our testbed we have about the same number of nodes over one-third of the length, increasing the number of collisions and retransmissions. We analyze this factor with dedicated experiments in Section 8.2.

The data from the WSN is collected by 2 gateways, installed on the same wall at 2 m and 80 m from the entrance. These test the effectiveness of our techniques for dividing the load of data collection and enabling one gateway to take over when the other fails. The gateways are connected to the PLC via Ethernet and powered by cables run from the tunnel power panel. A WiFi bridge at the entrance connects the PLC with a SCADA in our labs, allowing remote configuration of the experiments and collection of results.

8. EVALUATION

We evaluate our system first from the point of view of the application, assessing the ability to effectively and accurately close the control loop based on the data sensed by the WSN. Then, we look at the performance of the WSN itself.

8.1 Closing the Control Loop

First, we evaluate the response to *artificial* step-wise changes to the reference, to verify stability and convergence. Second, we evaluate the complete system according to its intended operation, with the reference properly set based on *real-world* light conditions.

Before these tests, we verified that the light readings of our sensors are indeed accurate in the tunnel, by comparing them against the illuminance probe we used for calibration in Section 6.2. Finally, recall that, as mentioned in Section 7, our testbed is realized by *partially* replacing the pre-existing lighting infrastructure. The latter includes old and unreliable lamps we do not control, influencing (and sometimes interfering with) the operation of our system.

Step response. The response of a closed loop system to step-wise changes in the reference point is fundamental to assess both the stability of the algorithm and its ability to track the reference. Moreover, for implementation reasons, the PLC changes the reference point of all sensors in small steps and not continuously. We ran the step-response tests at night, to avoid the bias induced by daylight, thus obtaining a controlled experiment in a real-world deployment.

Figure 8 focuses on two nodes, showing their target reference value (dashed line) and the light value actually sensed (solid line). The node position relative to the lamps bears a great influence. Node 4 is in an unfortunate place, receiving only little light from controlled lamps and a copious amount from uncontrolled ones, some of which are old and flicker. This situation is reflected in the noisy measures and imprecise convergence shown in the figure, still the system is able to track the step-wise reference variations. The position of node 7 is instead closer to what lighting design suggests, and its tracking of the reference is very good. The behavior of the other nodes is closer to node 7 than node 4.

In our final tunnel deployment, node placement follows from an accurate lighting design: we expect the performance to be similar to or better than the one of node 7, due to newer lighting equipment. However, the impossibility to fully redesign the lighting infrastructure of our testbed led to an interesting (albeit involuntary) worst-case experiment. Indeed, the results for node 4 confirm that, even with noisy measurements and an incorrect sensor placement, our system is robust enough to follow the reference trend.

Real-world reference. Figure 9 shows the results of experiments over 4 days. For convenience we group sensors in zones, roughly corresponding to the entrance (nodes 1–6), transition (7–12), and interior zone (13–15). In each chart, the solid line represents the light measured in the zone, while the dashed line is the reference. The dotted line in Figures 9(b) and 9(c) is the percent error between the two, whose scale is shown on the right-hand side y axis.

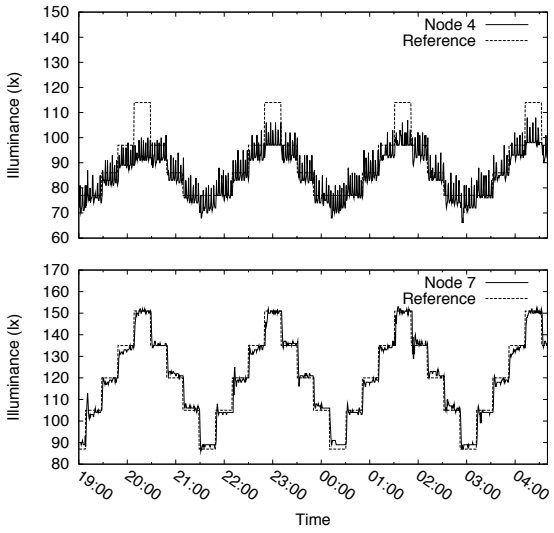


Figure 8: Evaluating the step response.

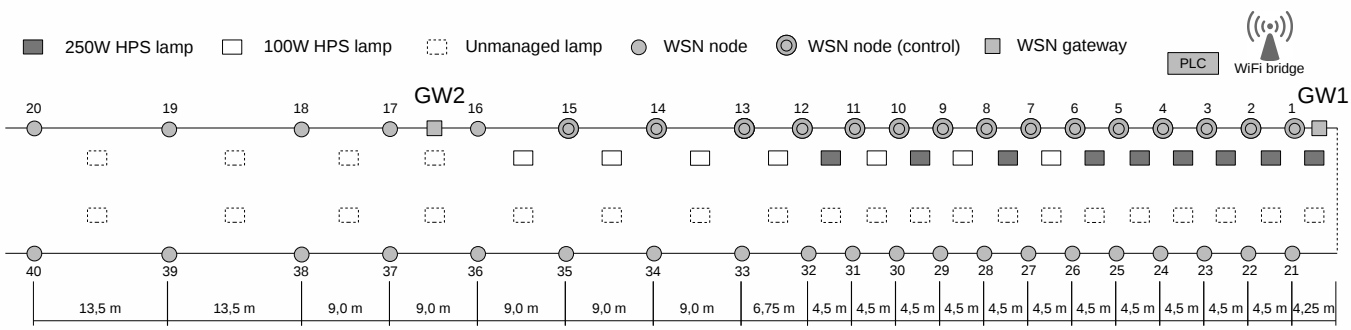
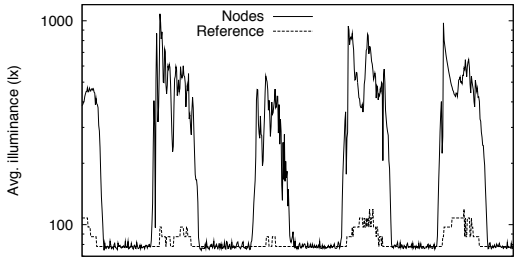


Figure 7: The equipment deployed in our testbed tunnel.

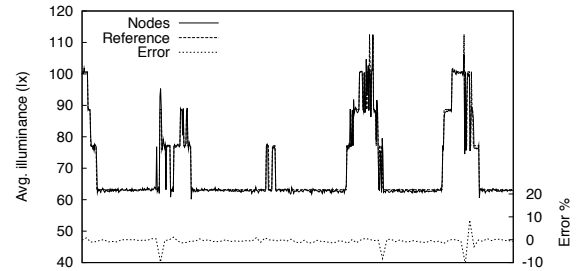
The dynamic range of any control system is limited by the actuators' capability. Due to the presence of a single power circuit, the actuation at the entrance of a tunnel is limited to a maximum of 150 lx. This prevents correct control in the entrance zone, shown in Figure 9(a). External light enters the initial part of the tunnel and the luminous flow achievable by the installed lamps is insufficient to match it—note the log-scale of the y axis. The reference set at night is instead achieved with precision.

Figure 9(b) shows the situation in the transition zone: node readings follow the reference so closely that the two are almost indistinguishable. Finally, as shown in Figure 9(c), in the interior zone the system matches the reference closely during the day, but remains slightly below it at night. This is due to node positioning that is not the result of an appropriate lighting engineering study. However, the error remains within $\pm 10\%$ of the reference.

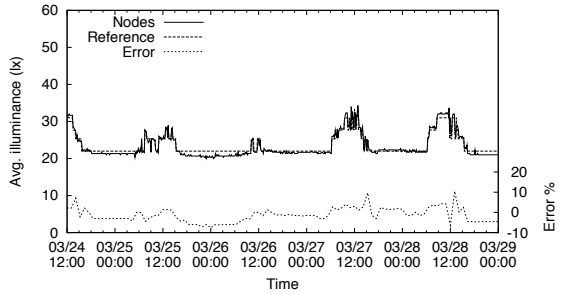
Despite the testbed limitations, these results show that the system can adapt effectively to the tunnel conditions.



(a) nodes 1-6: entrance zone



(b) nodes 7-12: transition zone



(c) nodes 13-15: interior zone

Figure 9: Performance of control in the testbed tunnel.

8.2 WSN Performance

We report about experiments over a 7-month period from August 13th, 2009 through February 2010. We initially separated the nodes in two networks on different radio channels: one for testing the control algorithm, and the other for testing routing and sampling. In mid-October, all 40 nodes became part of the same network, reporting to gateway GW2. We added the second gateway GW1 in mid-January. Apart from these major interventions, we performed other maintenance, e.g., to modify the software on the nodes. However, we never changed the batteries: Figure 10 shows energy consumption at select nodes, over the entire 7-month period.

As we discussed in Section 6.3, the control algorithm relies on different sampling rates along the tunnel. Nevertheless, we configured all nodes to report at the highest one (every 30 s), regardless of their position. This yields more data to test the control algorithm, and allows us to analyze the WSN behavior in more challenging conditions w.r.t. the final deployment. Besides light samples, each node reports once per minute other data made available through the SCADA (i.e., battery voltage, temperature, and routing parent) or used for debugging and experimentation purposes.

The WSN in our testbed is much denser, and challenging, than our target deployment, as we already noted. Therefore, at the end of this section we also report about experiments in a sparser deployment matching more closely our final one.

Data yield. A fundamental metric to analyze the performance of our routing layer is the amount of data correctly received from the WSN. Our application imposes a significant workload: the required reporting frequency for light samples results in an aggregated goodput (i.e., application messages collectively flowing in the WSN) of 1.3 msg/s, that increases to 2 msg/s if one includes also the reporting of system information. Nevertheless, the loss rate typically remains between 0.1% and 0.2%, as shown with a logarithmic scale in Figure 11. The spike on January 27th is caused by a 2-hour intervention required to update the gateways' software due to failures of the local connection to the sink. The other major spikes up to 10% are due to other transient errors on this connection, to forced shut-down of one of the gateways to test our redundancy mechanisms, and to minor maintenance to individual nodes. The remaining, smaller, variations are actual data losses in the WSN.

The reliability of communication is influenced also by the configuration of the underlying MAC layer, in our case the low-power listening (LPL) MAC available in TinyOS [28]. Prior to February 19th the MAC was configured with a sleep interval of 100 ms. We then changed it, as shown in Figure 11, initially to 250 ms and, on

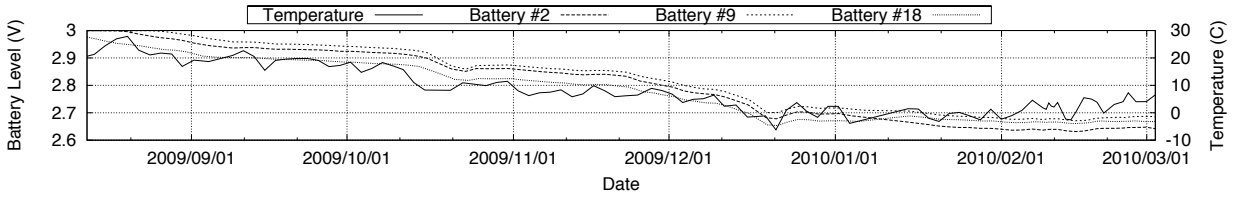


Figure 10: Temperature and battery levels on sample nodes over a 7-month period.

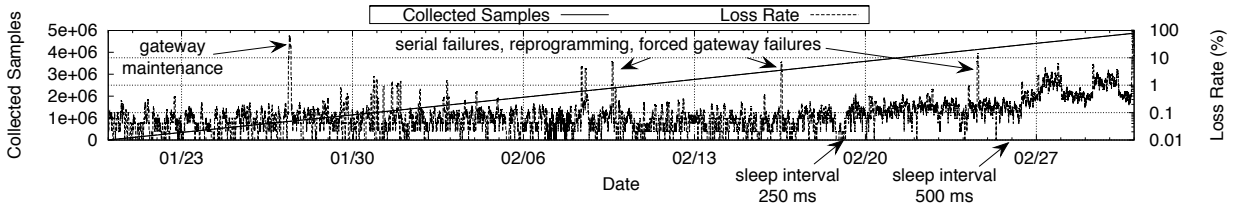


Figure 11: Total samples collected and loss rate over 1.5 months. The impact of the MAC sleep interval was also tested.

February 26th, to 500 ms. The increase to 250 ms results in a slight, still acceptable increase in loss rate. Instead, the 500 ms interval appears to be incompatible with our high throughput and network density, as the time spent transmitting and waiting for the receiver to wake up becomes significant. Apart from these experiments, the rest of the 7-month period used the 100 ms sleep interval.

Timely delivery. In a closed-loop system, high data yield alone is not sufficient. For data to be useful, it must arrive *on time*. In our system, the control algorithm runs every 30 s on the data collected during that interval. Each data sample reported by a node is timestamped at the gateway, allowing the PLC to recognize stale data. The jitter between samples from the same node is therefore of paramount importance, as shown in Figure 12. If two samples from the same node are received more than 30 s apart, the PLC *may* execute one of its cycles without a sample from the node, as in the case of node A on the right-hand side of Figure 12. However, if two samples are more than 60 s apart, as in the case of node B, the PLC *will* miss a sample from one or more intervals.

The critical interval is therefore between 30 s and 60 s. Figure 13 shows the cumulative distribution function of the sample reporting jitter in this interval, for the same LPL settings considered earlier, and for the sparse network described at the end of this section. As expected, the largest 500 ms sleep interval generates an excessive jitter: because of packet losses, about 3.5% of the samples miss the 30 s deadline, and a small fraction ($<$0.5%) misses the 60 s one. When using the smallest 100 ms sleep interval, 1.5% of the samples misses the 30 s deadline. Increasing the sleep interval to 250 ms introduces an additional 0.5% loss. However, in both cases the system recovers the situation within 60 s. Therefore, it is never the case that the PLC misses a sample from the same node for two or more consecutive intervals. This performance, achieved without routing mechanisms devoted to reducing jitter, is perfectly in line with the requirements of our control problem: the only consequence of these delays is a minor increase in convergence time.

Resilience to gateway failures. As our closed-loop control system must guarantee continuous operation, the WSN must automatically recover from failures, and limit their effects. In our target scenario, the WSN will be sparser than our testbed but still dense enough to allow alternate routes in the presence of one or more node failures, as we describe at the end of this section. The worst-case scenario is instead failure of one of the gateways, as this would prevent delivery to the PLC of *all* the data funneled through the failed sink. To reproduce this situation, we remotely forced the sink to disable its radio, disconnecting the gateway from the WSN. In this experiment, we killed gateway GW2, restored it after 2.5 hours, then killed the other gateway GW1. The two steps in the top chart of Figure 14 show the increase in data loss when either gateway fails. The failure of GW2 is more disruptive, as it collects data from more nodes, due to its position. After each failure, the cumulative loss rate decreases, and eventually converges to the previous values. As expected, losses are induced by gateway failures only: as shown in Figure 14, restoring GW2 did not affect the loss rate.

The bottom of Figure 14 shows instead that jitter increases sharply in the presence of a gateway failure. The amplitude of the peak corresponds to the time required by the nodes to switch to the new gateway. This occurs when the next tree refresh message is received: as one of the gateways is missing, all nodes receive the message only from the remaining gateway, and select their parent

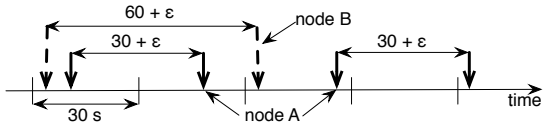


Figure 12: Impact of delays on the control algorithm. Vertical arrows denote arrival of a sample at the PLC.

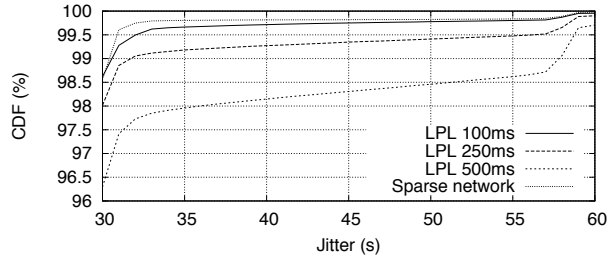


Figure 13: CDF for the sample reporting jitter.

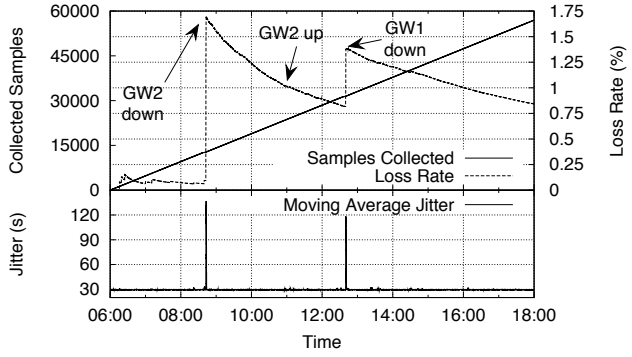


Figure 14: Forced gateway failures.

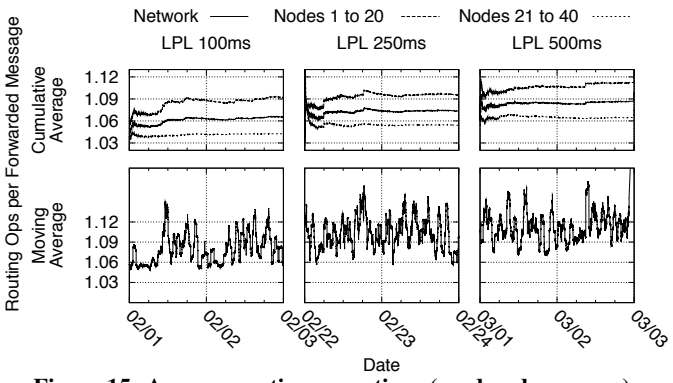
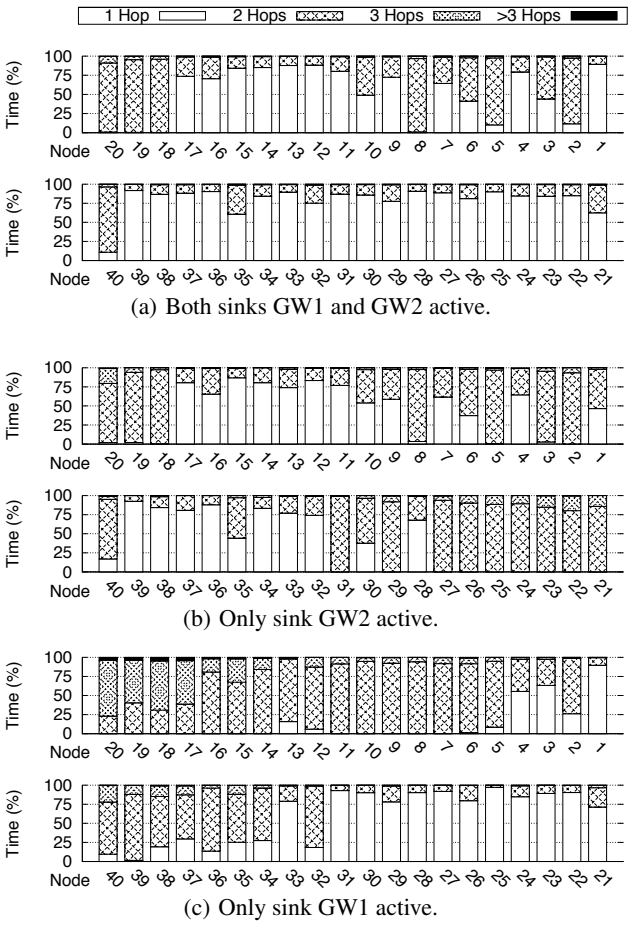


Figure 15: Average routing operations (send and recovery) per message forwarded per node.

accordingly. In our case, a tree refresh message is sent every 3 minutes: therefore, the worst-case delay is twice this time.

A closer look at routing. We obtain a high data yield thanks to mechanisms that overcome communication failures, i.e., retransmissions of non-acknowledged data and recoveries from the child cache when missing data is detected. Figure 15 shows the average number of *operations* issued by the routing protocol per message successfully forwarded by each node to its parent. In an ideal network, this value is 1. The chart shows the results with the usual LPL sleep intervals, each during a 2-day experiment. An operation in this context is the sending of a data tuple, or its recovery due to link failure. This high-level cost does not consider the operation details, e.g., the messages actually required to recover the lost data from the child cache, or the MAC overhead due to competing on the wireless medium. Nevertheless, it offers an indication of the effort made by the routing layer to sustain the high delivery rates in our dense setting. We provide a more detailed estimate of lifetime next. The top, cumulative average plots show that the cost to the whole network (the middle line in each plot) increases from approximately 6% for the smallest sleep interval to 10% for the largest. The bottom plots show the actual data as a moving average. For the smallest interval of 100 ms there is a higher cost during the day, when vehicular traffic slightly interferes with communication.



(a) Both sinks GW1 and GW2 active.

(b) Only sink GW2 active.

(c) Only sink GW1 active.

Figure 16: Percentage of time spent by each node at a given distance (hop count) from a sink.

Additional insights can be gathered by considering separately the costs of nodes on opposite walls. Figure 15 shows that the costs for nodes on the same wall as the sinks (nodes 1-20) are approximately 5% higher than for nodes on the opposite wall. To understand why, we compute the fraction of time a node spends at a given hop count from the sink. Figure 16(a) shows that nearly all nodes that spend more than 50% of their time at 2 hops are located on the same wall as the sinks. Indeed, nodes communicate better with nodes on the facing wall: they select one of these as their parent, which then crosses the tunnel back using a second, high quality link to the sink. Although a multi-hop path costs more than a 1-hop link, the extremely low cost observed for 1-hop links to sinks can be attributed to the fact that these do not duty-cycle like all other nodes, therefore transmissions towards them have very low failure rates.

With this in mind, we note that when both gateways are active, nodes always select the one with the shortest path. This is clearly seen in the other two plots in Figure 16, corresponding to the gateway failure experiments of Figure 14. Interestingly, a comparison among these charts shows that Figure 16(a) (both gateways active) appears to be the "union" of Figure 16(b) and 16(c) (only one gateway active): a node behaves in the WSN with both gateways as in the best of the configurations with only one gateway active.

Expected lifetime. It is generally difficult to predict the lifetime of WSN nodes. This quantity is indeed affected by many unpredictable aspects, including the effect of environmental conditions on battery performance—an important factor in our target scenario.

To obtain a lifetime estimate, we equipped 6 nodes evenly distributed along the testbed with new batteries, and recorded their voltage readings using the on-board sensor during 22 days of continuous operation. We determine the expected lifetime as follows:

1. For each day of our experiments, we match the day-long average voltage and average temperature against the battery discharge profile we obtained from the manufacturer [10]. This determines the service hours provided by the battery, given the current voltage and temperature.
2. We compute the average temperature for every day in 2009, based on publicly-available temperature data gathered by a weather station close to the testbed.
3. Using the temperature data at point 2, we replicate "in the future" (i.e., beyond the experiment duration) the battery discharge behavior we observed, essentially simulating the latter until the number of available service hours reaches zero.

The procedure above greatly *underestimates* lifetime. First, battery discharge profiles depend on the discharge current. In point 1 above, we use the profile for a 100 mA average discharge current, the lowest value in our battery data sheet. However, the current for WSN nodes running a LPL-like MAC protocol in configurations similar to ours [22] is expected to be a few mA. As shown in Figure 17 for our batteries, an order of magnitude difference in discharge current determines a significant increase of service hours.

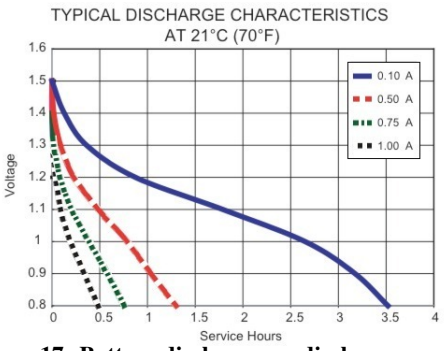


Figure 17: Battery discharge vs. discharge current.

Second, as we replicate the behavior observed at the beginning of a battery's life, we are considering the portion of the discharge profile where the battery loses service hours more rapidly.

Figure 18 illustrates the results of our analysis according to the LPL configuration. The minimal requirement of 1-year lifetime we stated in the introduction is always satisfied. The best performance always corresponds to a 250 ms sleep interval, the best trade-off between the power consumed in channel checks and packet strobing during transmissions. Instead, running LPL with a 500 ms sleep interval yields the worst performance in most cases. We conjecture that in this setting the power consumption due to long strobing outweighs the gains yielded by less frequent channel checks. Node 31, on the other hand, performs worse with a 100 ms sleep interval. Presumably, its location in the topology diminishes the bad effects of the 500 ms setting, which provides slightly longer lifetime than the 100 ms one. Instead, node 20 shows a markedly higher expected lifetime in all settings: this node is frequently a leaf in the routing tree, and therefore experiences a reduced routing load.

Towards the final deployment. The WSN behavior in our testbed is deeply affected by the high density of nodes. The final tunnel deployment will be much sparser. The placement of nodes is driven by considerations of the lighting engineers, who place the nodes in the interior zone—where fine-grained measurements are less important—with an inter-node distance up to 60 m. To gather insights on long-range links, we setup experiments using 8 new nodes, divided evenly between the two walls. The experiments ran from December 4th to the 14th, on a different radio channel w.r.t. the rest of the WSN in Figure 7. For these tests we relied on GW1 (recall that the two-gateway tests began in mid-January). The 8 nodes were positioned as follows. Starting from the sink on GW1, nodes on the same wall were placed 60 m apart, and approximately 30 m from the closest node on the facing wall. To investigate if connectivity was retained in the presence of node failures, we disconnected two of the central nodes (node 4 and 5) on December 10th and 11th, yielding an inter-node distance of 120 m on the same wall. These experiments ran with a 100 ms LPL interval.

Figure 19 shows the results. In the sparse WSN we setup, our routing achieves 99.98% delivery with an extra cost (i.e., #operations) less than 4%, as shown by Figure 19(a). Spikes in the moving average for cost are due to the selection of a low quality parent, and are absent in times of low vehicular traffic (e.g., the weekend of December 12-13). The overall improved performance w.r.t. previous experiments is motivated by the decreased density: channel contention is reduced and the MAC layer handles communication more effectively. Notably, the forced failure of the two center nodes has no impact, confirming the considerable length of reliable links in the tunnel environment. The routing tree, shown in Figure 19(b), has a higher depth w.r.t. the situation in Figure 16. Nevertheless, this does not negatively affect the jitter, as shown in Figure 13.

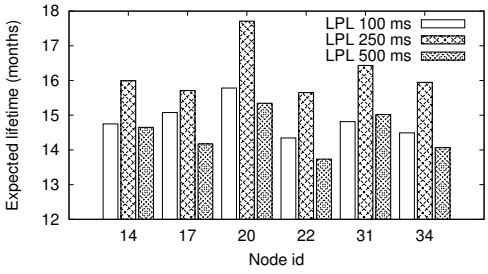
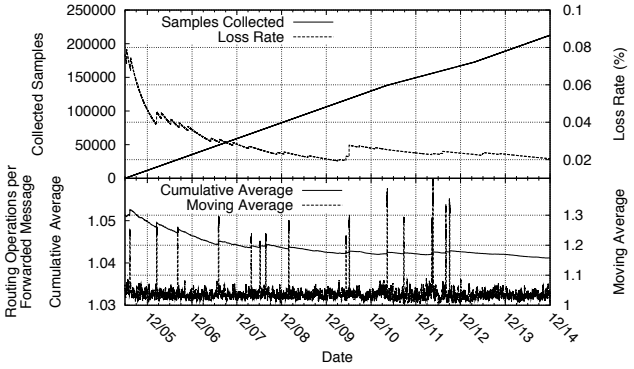
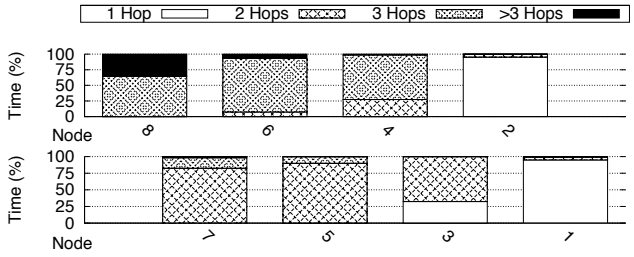


Figure 18: Expected lifetime, beyond one year.



(a) Data loss and cost in terms of number of operations



(b) Time spent by each node at a given distance from the sink

Figure 19: Experiments in a sparse setting.

Although the ultimate answer will come from the final tunnel deployment, these results show that the latter should perform better than the overly-dense experimental testbed reported in this paper.

9. BEYOND ADAPTIVE LIGHTING: FIRE DETECTION

As mentioned in the introduction, TRITon is a large project encompassing several technologies. At one point, another team tested their camera-based fire detection system with real fires staged by the local fire department. As this occurred in our testbed tunnel, we took advantage of the event to investigate whether the WSN we conceived for light sampling could be used also for fire detection.

Indeed, the ISL29004 illuminance sensor we used relies on two photodiodes: the first one (D_A) is sensitive to both visible and infrared (IR) light, while the second one (D_B) is sensitive mostly to IR. Measuring illuminance requires that the wavelength of incident light is compensated to follow the human eye response, which is achieved by "subtracting" the response of D_B from the one of D_A. It is well-known that fire, unlike tunnel lamps, emits a significant fraction of light in the IR spectrum. Detecting fire with the ISL29004 sensor then essentially means monitoring the output of D_B: in the presence of fire, this diode immediately reports a value much higher than D_A. Therefore, in practice, running this experiment came at negligible cost. We modified the sensor driver to report IR along with illuminance, and made minimal changes to

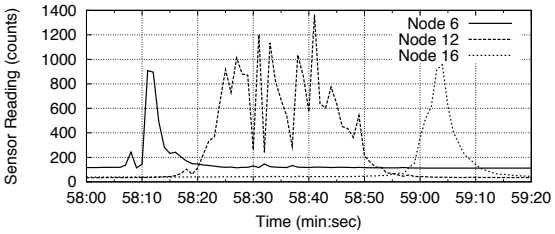


Figure 20: Detecting fire through infrared light sensing.

the sampling rate (1 s) and message format.

In the experiment we report here, a fireman on the back of a truck held a tube connected to a propane tank, which continuously fueled a flame at the free extremity of the tube. The truck moved from right to left w.r.t. Figure 7. Figure 20 shows the data reported by our WSN, charting over time the IR values from nodes at different distances inside the tunnel. The presence of a flame causes a distinct and instantaneous increase in the IR value. The many peaks at node 12 are due to the fireman waving the flame in front of it.

Full-fledged fire detection requires more in-depth studies. However, this impromptu experiment hints at the fact that, once a WSN is deployed in a tunnel, applications other than lighting become feasible, possibly with only minimal changes to the base design.

10. CONCLUSIONS AND FUTURE WORK

We reported on a WSN-based system for adaptive, closed-loop control of lighting in road tunnels. Adaptive control is expected to increase the safety of tunnels and reduce their power consumption. The system is designed for long-term, real-world operation: the WSN is fully integrated with industrial-strength tunnel automation equipment. The system is running in an operational tunnel serving as an experimental testbed. Several months of experiments proved that our WSN hw/sw design enables closed-loop control in the harsh tunnel environment, matching the stricter requirements posed by actuation w.r.t. common monitoring-only deployments.

The main focus of our immediate activities is the final deployment, which targets a longer tunnel with a higher volume of traffic, and includes 4 gateways and 88 nodes divided evenly among the two carriageways. At the time of writing, the installation of the LED and HPS lamps is being finalized, after which automation and WSN equipment will follow. Real-world operation in a tunnel whose infrastructure fully supports our solution will enable us to concretely demonstrate the advantages of WSN-based adaptive lighting, e.g., the savings in the tunnel power consumption. However, the availability of a reliable WSN infrastructure paves the road for applications beyond adaptive lighting. We already verified the feasibility of fire detection; pollution (CO) monitoring and in-network actuation are now being designed.

Acknowledgments. The Autonomous Province of Trentino partially funded the project and granted us access to several operational tunnels through its Dept. of Road Management (Servizio Gestione Strade). The local fire department kindly supported us during the tests reported in Section 9. The activities reported here have also been partially funded through the EU Cooperating Objects Network of Excellence (CONET—FP7-2007-2-224053).

11. REFERENCES

[1] M. Ceriotti et al. Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment. In *Proc. of the 8^{th} Int. Conf. on Information Processing in Sens. Netw. (IPSN)*, 2009.

[2] S. Cheekiralla. Wireless sensor network-based tunnel monitoring. In *Proc. of the RealWSN Workshop*, 2005.

[3] CIE—International Commission on Illumination. *Guide for the Lighting of Road Tunnels and Underpasses (CIE 88-2004)*, 2004.

[4] P. Costa et al. The RUNES middleware for networked embedded systems and its application in a disaster management scenario. In *Proc. of the 5^{th} Int. Conf. on Pervasive Computing and Communication (PerCom)*, 2007.

[5] P. Costa, L. Mottola, A. L. Murphy, and G. P. Picco. Programming wireless sensor networks with the TeenyLIME middleware. In *Proc. of the 8^{th} ACM/USENIX Int. Middleware Conf.*, 2007.

[6] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[7] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. The collection tree protocol. In *Proc. of the 7^{th} Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 1–14. ACM, 2009.

[8] Q. Han, P. Anura Jayasumana, T. Illangasekare, and T. Sakaki. A wireless sensor network-based closed-loop system for subsurface contaminant monitoring. In *Proc. of the 1^{st} NSF Workshop on Next Generation Software*, 2008.

[9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proc. of the 9^{th} Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, pages 93–104. ACM, 2000.

[10] W. X. Hu, 2010. Duracell Corp - Private correspondence.

[11] Y. Kim, R. G. Evans, and W. M. Iversen. Evaluation of closed-loop site-specific irrigation with wireless sensor network. *Journal of Irrigation anf Drainage Engineering*, 135(1), 2009.

[12] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proc. of the 1^{st} Conf. on Networked Systems Design and Implementation (NSDI)*, 2004.

[13] M. Li and Y. Liu. Underground coal mine monitoring with wireless sensor networks. *ACM Trans. on Sensor Networks*, 5(2):1–29, 2009.

[14] W. Liu, J. Principe, and S. Haykin. *Kernel Adaptive Filtering: A Comprehensive Introduction*. John Wiley, 2010.

[15] L. Luo, T. F. Abdelzaher, T. He, and J. A. Stankovic. EnviroSuite: An environmentally immersive programming framework for sensor networks. *ACM Trans. on Embedded Computing Systems*, 5(3), 2006.

[16] J. Lynch et al. Implementation of a closed-loop structural control system using wireless sensor networks. *Structural Control and Health Monitoring*, 3(15), 2008.

[17] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of the 1^{st} Int. Workshop on Wireless Sensor Networks and Applications*, 2002.

[18] D3S Research Group. `teenylime.sourceforge.net`.

[19] J. Molina-Garcia-Pardo, M. Lienard, and P. Degauque. Propagation in tunnels: Experimental investigations and channel modeling in a wide frequency band for MIMO applications. *EURASIP Journal on Wireless Communications and Networking*, 2009(3), 2009.

[20] L. Mottola, G. Picco, M. Ceriotti, S. Guna, and A. Murphy. Not All Wireless Sensor Networks Are Created Equal: A Comparative Study On Tunnels. *ACM Trans. on Sensor Networks (TOSN)*, 7(2), 2010.

[21] H. Park, J. Burke, and M. Srivastava. Design and implementation of a wireless sensor network for intelligent light control. 2007.

[22] S. Park, A. Savvides, and M. Srivastava. Battery capacity measurement and analysis using lithium coin cell battery. In *Proc. of Int. Symp. on Low Power Electronics and Design (ISPLED)*, 2001.

[23] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proc. of the 5^{th} Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2005.

[24] V. Singhvi, A. Krause, C. Guestrin, J. Garrett, and H. Matthews Scott. Intelligent light control using sensor networks. In *Proc. of the 3^{rd} Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2005.

[25] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline. PIPENET: A Wireless Sensor Network for Pipeline Monitoring. In *Proc. of the 6^{th} Int. Conf. on Information Processing in Sensor Networks*, 2007.

[26] Z. Sun and I. Akyildiz. Channel modeling of wireless networks in tunnels. In *Proc. of GLOBECOM*, 2008.

[27] The u2010 Project. Final demostrator. `www.u-2010.net`.

[28] TinyOS. TEP 105—Low Power Listening. `www.tinyos.net`.

[29] TinyOS. TEP 119—Collection. `www.tinyos.net`.