Comparative Performance Analysis of the PermaDozer Protocol in Diverse Deployments

Matthias Keller, Matthias Woehrle[‡], Roman Lim, Jan Beutel, Lothar Thiele

ETH Zurich, Computer Engineering and Networks Laboratory, 8092 Zurich, Switzerland
Email: {kellmatt, lim, beutel, thiele}@tik.ee.ethz.ch
[‡]Delft University of Technology, Embedded Software Group, 2628 CD Delft, The Netherlands
Email: m.woehrle@tudelft.nl

***Abstract*—In this paper, we present a performance analysis of the communication stack of the PermaDozer application. Our offline analysis is based on long-term performance data from three diverse real-world wireless sensor network deployments. Two of the deployments considered are located in the Swiss mountains at 3.500 meters a.s.l., the third deployment is located along a river in the Swiss midlands. Apart from climatic differences, the three deployments also vary in network size, network density, node placement, and type and quantity of RF interference. From September 2008 to May 2011 more than 99.6 million WSN packets have been collected serving as the dataset.**

All three deployments are based on the same software and hardware. This allows us to comparatively study the performance of the PermaDozer protocol under different deployment settings and environmental conditions. For the period between June 2010 and May 2011, our networks achieved a data yield of $\geq 99.5\%$. But, we can also clearly notice that achieving this performance requires varying effort in terms of radio duty cycle resulting in different power consumption and network lifetime.

***Index Terms*—Wireless Sensor Networks, Performance Evaluation, Long-Term Deployments**

I. Introduction

The PermaSense project [7] collects long-term measurements of diverse physical phenomena for geophysical research. For PermaSense, the PermaDAQ [2] platform is used in several different deployments monitoring geophysical properties like surface temperature profiles, crack movement etc. The PermaDAQ platform is using TinyNodes [6] nodes running PermaDozer, a software built on top of TinyOS [10] and based on the Dozer [3] ultra low-power network protocol. While the majority of sensor nodes is deployed for sensing geophysical phenomena, nodes are also used for monitoring and controlling other devices, *i.e.*, more powerful embedded computers and networked cameras.

The performance of real-world deployments is often affected by phenomena that are specific to the deployment location and cannot be covered by testbed experiments or simulation. Once a decrease in certain performance metrics, *i.e.*, data yield, network performance or node lifetime, is detected, it is usually not easily understood. In this paper, we want to contribute to this problem by comparing the performance of a fixed system architecture at three different deployment locations. The purpose of this study is firstly to learn about the characteristics of different environments, secondly to quantify how the systems reacts to different environments, and thirdly to learn how we can detect performance changes and design an online health monitoring infrastructure.

The PermaDozer application includes health monitoring that periodically logs vital system information in-band with sensor data to the PermaSense data repository[1]. In this work, we utilize the collected health-monitoring data to characterize system operation. In particular, we study three different deployments at the Matterhorn, the Jungfraujoch and river Thur and characterize their system operation. Each deployment differs in scope and in deployment size. More importantly, each deployment is embedded into an environment that seemingly impacts the system operation of the PermaDozer protocol. Based on the characterization of the deployments, we present a comparative, in-depth analysis of the deployments to identify the impact of the environmental conditions on the common sensor network platform used.

This paper provides the following contributions:

- We describe a health monitoring infrastructure that allows us to characterize system operation.
- We use the health data collected from several long-term deployments to perform diverse analyses that allow us to get insight into the performance of the specific deployments. Considered essential metrics are data yield, duty cycle, packet duplications, packet delay and node resets.
- We comparatively study our three different deployments running the same node hardware and software. Thereby, we can show how the differing deployment characteristics and physical environments impact system operation.

II. Case Study Deployments

In this section, we describe and characterize the three deployments considered and as such we provide a foundation for the interpretation and comparison of measurements presented in sections V and VI.

A. Location

The locations of the Matterhorn, Jungfraujoch and Thur deployments are shown in Figure 1.

[1]Access the data online at http://data.permasense.ch.

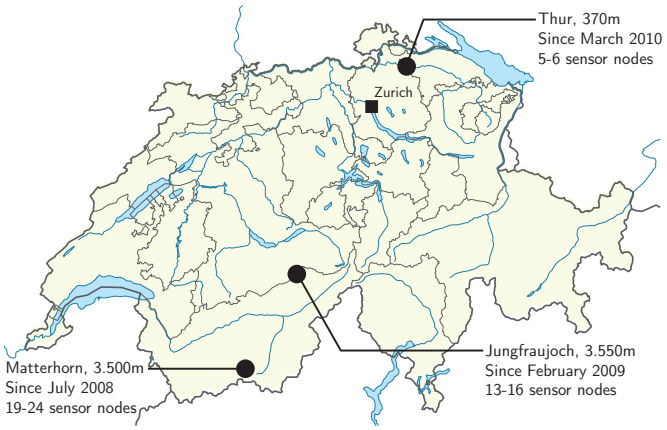


Figure 1. PermaSense deployment locations in Switzerland.

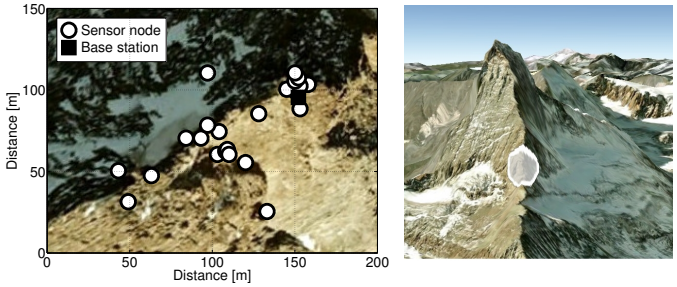


Figure 2. Sensor nodes form a single densely populated patch in mostly line-of-sight conditions along the Matterhorn Hoernli ridge. (Images: © 2010 Google)

Two of our three deployment sites are located in the Swiss Alps at an altitude of 3.500 meters above sea level in remote glacier mountain environments: The Matterhorn deployment site is located at the Hoernli ridge of the Matterhorn. The Jungfraujoch deployment site is co-located with the Jungfraujoch railway station and the High Altitude Research Station Jungfraujoch. In contrast, the Thur deployment is located along the Thur river at an altitude of 375 meters above sea level in a rural and agricultural setting.

B. Network Size

The largest installation is located on the Matterhorn, and consists of up to 24 nodes. Up to 16 nodes are installed at Jungfraujoch. The Thur deployment consists of up to six sensor nodes.

C. Node Placement

The node placement at the Matterhorn deployment is shown in Figure 2. Most of the sensor nodes are either in line-of-sight with the base station, or in line-of-sight with another sensor node, that in turn is in line-of-sight with the base station.

Figure 3 shows the sensor node placement at Jungfraujoch where two disjoint sensor node patches are located at the North and South faces of Jungfraujoch and the base station is situated on top of the building. Based on the long distance between the two patches and the obstructed line-of-sight, we assume that nodes belonging to different patches are not able to communicate nor interfere with each other directly.

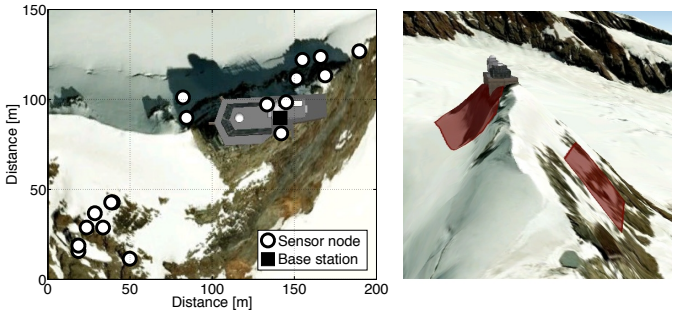


Figure 3. The Jungfraujoch field site consists of two disjoint (non-line-of-sight) sensor node patches. The base station as well as two relay nodes are located on the roof of the Sphinx building. (Images: © 2010 Google)

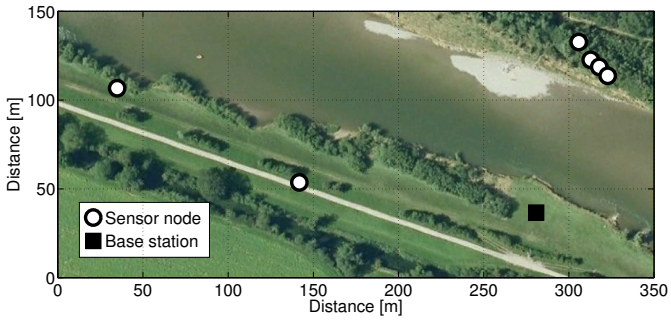


Figure 4. Nodes are located on both sides of the Thur river. All sensor nodes are mast mounted at an elevation of 1.5 meters. (Image: © 2010 Google)

In contrast to the two deployments in the mountains, the sensor nodes at the Thur deployment are situated on a river bank. The sensor node placement is shown in Figure 4, here, all sensor nodes are in line-of-sight with each other.

The PermaSense deployments are exposed to extreme outdoor environments with harsh conditions. Figure 5 shows the mean air temperatures for the three deployment sites. At the high-alpine sites Matterhorn and Jungfraujoch, the mean annual air temperature is typically below 0°C documenting the extremely cold environment. In contrast, the monthly mean air temperature is above 0°C for 11 months at the Thur deployment site. The annual variability of ≈14°C at Matterhorn and Jungfraujoch, are somewhat less than the 19.0°C at the Thur deployment location. The daily temperature variation that is dominated by the impact of direct exposure to the sun and low night time temperatures is considerably higher, typically on the order of 40-50°C for all locations.

The sensor nodes at Matterhorn and Jungfraujoch are situated in steep bedrock and mounted directly on the rock surface. Depending on the morphology, the exposition to wind and a local variability in precipitation, snow and ice accumulates on a large part of these deployments. While at the Matterhorn location there is only little snow observed in winter and there remains almost no snow during summer, the snow cover around Jungfraujoch is significant. Especially on the north side

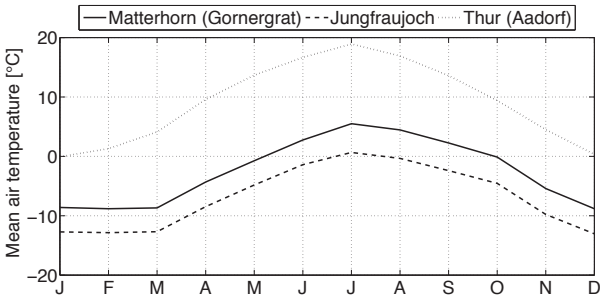


Figure 5. Mean air temperatures recorded between 2005 and 2011 at the closest meteo stations next to the deployment sites. (Source: SwissMeteo)

a large part also remains during summer, partly transforming into ice resulting in (*i*) an attenuation of radio signals and (*ii*) of fast temperature changes. At the Thur the seasonal snow cover is negligible as all nodes are mast-mounted.

D. Potential RF Interference

Apart from our own site infrastructure that uses 2.4 GHz and 5 GHz WiFi signals, we are not aware of any particular local RF interferers at the Matterhorn and Thur deployment sites. The Matterhorn deployment site is passed by climbers who are on the way to the summit during summer, however, both locations are only rarely visited by humans.

There are three facts that might result in remarkable RF interference at Jungfraujoch. First, Jungfraujoch is a very popular tourist location with up to 8.000 visitors per day. There is a high usage of GSM communication in the area. Secondly, the research station at Jungfraujoch is hosting several scientific and commercial stations for measuring geophysical, meteorological and astrophysical properties. We can not exclude the possibility of emissions from co-located experiments. Thirdly, the area is surrounded by several transmissions stations used for broadcasting strong signals, *i.e.*, GSM or TV broadcast. For example, the federal office for communication (BAKOM) lists ten distinct GSM/UMTS base stations in the vicinity of less than 200 meters around the deployment site.

III. Environmental Monitoring with PermaDozer

Based on TinyOS, the PermaDozer software integrates the Dozer [3] protocol, a library of routines for accessing different types of sensors, a driver for accessing an external storage, and a file system implementation. PermaDozer is a building block of the PermaDAQ [2] platform that defines the system architecture including special-purpose sensing hardware and mechanics.

We have to distinguish between two major versions of the PermaDozer software and hardware. While the initial version was based on TinyOS 1 and only supporting TinyNode 584 sensor nodes, the current version supports TinyNode 184 [13] nodes based on TinyOS 2. Both platforms are based on a MSP430 micro controller, but, the Semtech XE1205 radio found on the older TinyNode 584 was replaced by a Semtech SX1211 [12] radio chip in the newer TinyNode 184 operating in the 868 MHz Band.

Apart from changes in the operating system and hardware platform, further features such as the implementation of an event log mechanism or the possibility to measure the RSSI (received signal strength indicator) are only available in the second generation of PermaDozer.

In the following description, we will only focus on the recent version of PermaDozer that is now in use at all three deployments: Existing legacy sensor nodes at Jungfraujoch and Matterhorn were replaced in January and June 2010, respectively. Sensor nodes at the Thur deployment were equipped with newest generation hardware and software since the beginning in March 2010.

A. System Model

Starting from June 2010, the event log mechanism of the PermaDozer software is also used to record previously hidden states of the internal state machine of the Dozer protocol. While it was previously only possible to detect parent switches by tracking periodically transmitted parent information, the new solution offers two advantages: (*i*) Instead of only periodically sampling a state, it is now possible to reconstruct the order of actions from event traces. (*ii*) It provides further insights, *i.e.*, not only stating a topology change, but also reasons why the protocol decided to switch the parent.

As a prerequisite to understand the analysis of event log data presented in the remainder, we now provide a behavior model of the Dozer protocol stack. While we refer to the original paper [3] of Burri *et al.* for a complete description of the Dozer protocol, we will now only describe mechanisms that are important for our performance analysis. The finite-state machines presented for describing the protocol operation have been derived from the actual implementation. For the ease of reading, we graphically distinguish between observable and non-observable state transitions. While non-observable transitions are shown with solid lines, *i.e.*, *No connection* in Figure 6, observable state transitions are shown with dashed lines, *i.e.*, *Activity detected (AD)* in Figure 6.

In Dozer [3], sensor nodes communicate on a dynamic tree topology that originates from a sink[2]. The tree topology is constrained to 1:n one-hop parent-child relationships; the parent node has a shorter distance to the sink than its children. The sink is periodically initiating communication by sending beacon messages for announcing the network and requesting its child nodes to send data. A child node receiving a beacon not only tries to upload its data to the parent, but also generates and transmits another beacon message. This way, sink-initiated beacons are transmitted across the whole multi-hop network. Coordinated, ideally exclusive communication between parent-child pairs firstly reduces the risk of collisions, and secondly enables scheduled wake-ups that allow to run the radio on a very low duty cycle.

A top-level description of the Dozer protocol is given in Figure 6. After hardware initialization, a node firstly snoops on

[2]The sink function is provided by the base station. Both terms are interchangeable in the scope of this paper.

the radio channel for activity detection. As soon as activity is detected, the node listens again on the channel for overhearing beacon messages of neighboring nodes, see Figure 7. Found potential parents are rated based on a utility function, the ordered list is traversed until the node is either connected to a parent or must return to the *Snoop* state, see Figure 8. The state machine for *Normal* operation is given in Figure 9.

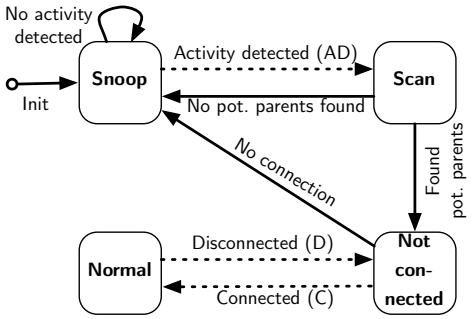


Figure 6. Top-level description of the Dozer protocol. Dashed lines refer to observable state transitions that are logged. Detailed explanations of the *Scan*, *Not connected* and *Normal* states are given in Figure 7, Figure 8, and Figure 9, respectively.

For the following section we assume a disconnected sensor node $\mathcal{N}$ operating in *Scan* state after detecting channel activity.

Parent discovery (*Scan*). Sink and sensor nodes in the *Normal* state announce the network by periodically sending beacon messages. Ideally, those announcements do not interfere. The current implementation uses a beacon interval of 30 seconds, thus node $\mathcal{N}$ listens for 30 seconds to receive beacons from as much different parents as possible. Received beacons and their reception time are stored in a list of potential parents. The reception time is important for future energy-efficient communication with a parent. If at least one beacon was overheard during the 30 second long scan, $\mathcal{N}$ changes into the *Not connected* state. If no beacons were received, $\mathcal{N}$ returns into the *Snoop* state.

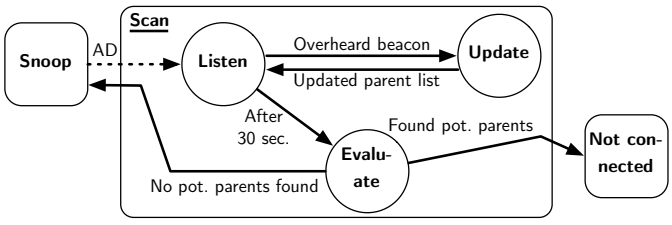


Figure 7. Simplified state machine for parent discovery. The sensor node tries to receive beacons from as many potential parents as possible during a 30 second long scan.

Network join (*Not connected*). Potential parents found are evaluated by a utility function that tries to minimize the resulting distance to the sink while ensuring that a parent with enough capacity for additional children and a good link quality is chosen. After ordering potential parents by the result of the utility function, the best potential parent is chosen. Ideally, after receiving another beacon from this parent, $\mathcal{N}$ sends a connection request, which is in turn acknowledged by a handshake from the parent node. Node $\mathcal{N}$ is connected as soon as the handshake is received, the following top-level state is *Normal*. However, handshake and beacon packets from a potential parent can get lost in practice. Thus, node $\mathcal{N}$ has to move on in the list of potential parents if either two beacon messages or a handshake message from a chosen parent are lost. If connecting to all potential parents fails, node $\mathcal{N}$ must go back into the *Snoop* state.

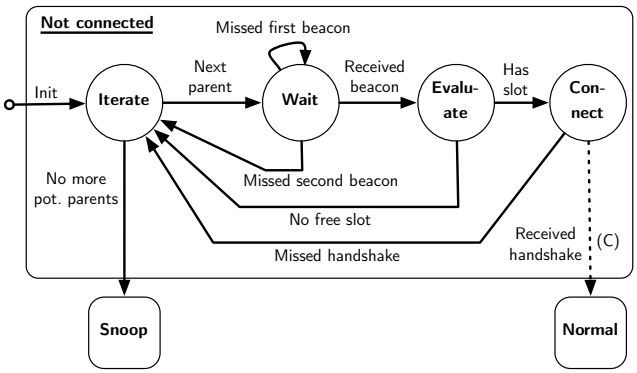


Figure 8. Simplified state machine of network join. The node is waiting for the beacon of a potential parent. The expected reception time is calculated from the reception time of previously received beacons. The node traverses through the list of potential parents until it successfully connects or gives up.

***Normal* operation.** In this state, node $\mathcal{N}$ acts both as a child and a parent. As this is not necessary for the scope of this paper, we limit this explanation to the view of a child and refer to [3] for the view of a parent. The periodically send beacon messages of the parent of $\mathcal{N}$ are synchronously received by all of its children. From the reception time and its own position in the children queue of the shared parent, each child calculates when it can upload data without interference with other nodes. In case of two consecutive beacon messages or message acknowledgements being lost, node $\mathcal{N}$ moves back to the *Not connected* state. It is then trying to connect to unused entries of the list of potential parents. Since this list was no longer updated since the last visit of the *Scan* state, there is a high probability of this list only containing stale (wake-up timing) information due to clock drift.

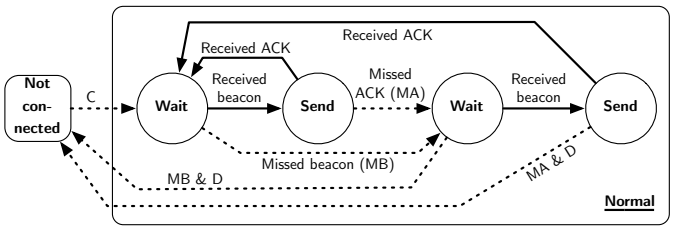


Figure 9. Simplified state machine of normal operation from the view of a network child. The node periodically wakes up to receive the beacon of its parent. If the beacon was received, the child node sends its data. Each packet is acknowledged separately, missed beacons or acknowledgements lead to the termination of the current try or to the node giving up the connection.

B. Data Acquisition

The collection of health information is handled by the data acquisition (DAQ) routine of PermaDozer. The following performance analysis exploits both attached packet application headers and dedicated health information data packets. In this section, we describe involved application headers and available health information.

Independent of the state of the Dozer protocol, the data acquisition routine is executed every 120 seconds. While there are a number of packet types, *e.g.*, packets containing sensor or node health information, that are generated periodically at each execution of the DAQ routine, there are also packet types, *e.g.*, a packet type for transmitting logged events, that are only sent on the occurrence of an event, *e.g.*, when the local event log queue is full.

After packet generation, the current value of the packet sequence counter is immediately added to the packet. The packet sequence counter is incremented after each generated packet or reset to zero if it reached the maximum number that is supported by the corresponding variable of fixed length.

Sensor nodes are neither aware of a global notion of time, nor are their clocks synchronized. Instead, the delay a packet experiences within the network is continuously updated until the packet is received by the base station that runs a network-synchronized clock. The time of packet generation is then reconstructed by subtracting the elapsed time on arrival from the current clock on arrival at the base station.

C. Packet Buffering

Harsh environmental conditions as found at mountain deployments can influence system operation. For instance, radio communication may not be possible for several weeks or even months if a sensor node is buried into ice and snow. Another example is the base station infrastructure that can only be operated if enough sunlight allows to recharge the batteries of the base station fast enough.

For mitigating the effects of disconnected operation and potential data loss, sensor nodes are equipped with 1 GB external flash storage. When temporarily disconnected from the sink node packets are written to the external storage immediately after generation. Upon re-connection all unsent packets backlogged are forwarded to the next hop as soon as the sensor node enters the *Normal* operation state.

IV. Long-term Performance Data

After operating real-world deployments for almost three years, we can access a large body of collected performance data. In the context of this paper, we are concerned about the following information that we will employ in the following performance analysis:

Node health information. A packet containing node health information is sent periodically every 120 seconds. The relevant information for the analysis presented contained therein is the node uptime, the network address of the current parent node, the number of child nodes, and the temperature reported by a Sensirion SHT11 sensor. The information captured always covers the state at the time of the data acquisition, i.e. the moment the data packet is generated, not when it is sent out. As an effect of this the recorded network address of the current parent node is thus only valid if the node is in the *Normal* operation state at the time of packet generation.

Packet delay. The packet delay is not only used for reconstructing the time of packet generation, it can also be used as a network performance metric.

Dozer events. Previously described *Missed beacon (MB)*, *Missed acknowledgement (MA)*, *Connected (C)* and *Disconnected (D)* events are also transmitted in-band and thus observable from outside. Additionally, nodes are reporting the occurrence of resets.

Link quality. On the reception of beacon or data packets, the received signal strength (RSSI) is measured. Measurements including the network addresses of the sending nodes are also transmitted in-band. This feature is only available on recent generation PermaDozer systems.

Table I
Analysis of network topology information that was recorded between June 2010 and May 2011

Deployment	Matterhorn	Jungfraujoch	Thur
# of sensor nodes	24	15	6
Avg. hop distance	1.7	1.9	1.2
Max. hop distance	7	7	5
In single-hop range of sink	21 of 24	8 of 15	6 of 6
Sink: Mean number of children	9.4	2.9	4.2

V. Performance Analysis

In this section, we evaluate the performance of the PermaDozer protocol in the presented three environments with different characteristics. As a prerequisite for this performance analysis, we apply a filter step [9] that firstly cleans the data set from packet duplicates, and secondly reconstructs the temporal order of packet generation to mitigate artifacts of clock inaccuracies.

A. Network Topology

A characterization of network topologies found is shown in Table I. Almost all sensor nodes are able to directly communicate with the sink at the Matterhorn network. But, PermaDozer is firstly limiting the number of children to at most 10 single-hop children per parent, and secondly adds a penalty to nodes with a high number of connected single-hop children when evaluating potential parent nodes during the parent selection process. As a result, at least 14 of 24 nodes are at least two hops away from the sink at all times.

Only half of the nodes are able to directly communicate with the sink at Jungfraujoch, packets must often be relayed over three or more hops. In contrast, data is usually relayed over at most two hops at the small network at Thur.

Table II
LINK CHARACTERISTICS BASED ON RSSI MEASUREMENTS FROM JUNE 2010 TO MAY 2011

Deployment	**Matterhorn**			**Jungfraujoch**			**Thur**		
# of samples	10,347,562			4,805,958			2,785,221		
Link class	**# of links**	$\bar{\sigma}_R$	$\max \sigma_R$	**# of links**	$\bar{\sigma}_R$	$\max \sigma_R$	**# of links**	$\bar{\sigma}_R$	$\max \sigma_R$
Bidirectional communication – downlink									
Good links (RSSI>-65 dBm)	143 (33.7%)	2.1	7.6	14 (20.6%)	6.3	9.4	20 (31.8%)	3.4	5.0
Med. links (-65≥RSSI>-85 dBm)	208 (49.1%)	2.9	16.6	40 (58.8%)	3.0	7.0	41 (65.1%)	3.2	9.9
Bad links (RSSI≤-85 dBm)	73 (17.2%)	1.4	4.3	14 (20.6%)	1.5	2.4	2 (3.2%)	2.2	2.8
Bidirectional communication – uplink									
Good links (RSSI>-65 dBm)	149 (35.1%)	2.1	7.3	14 (20.6%)	5.8	8.6	17 (27.0%)	3.5	5.6
Med. links (-65≥RSSI>-85 dBm)	201 (47.4%)	3.1	15.5	38 (55.9%)	3.0	6.7	44 (69.8%)	3.3	12.6
Bad links (RSSI≤-85 dBm)	74 (17.5%)	1.7	5.7	16 (23.5%)	1.9	2.5	2 (3.2%)	1.7	2.2

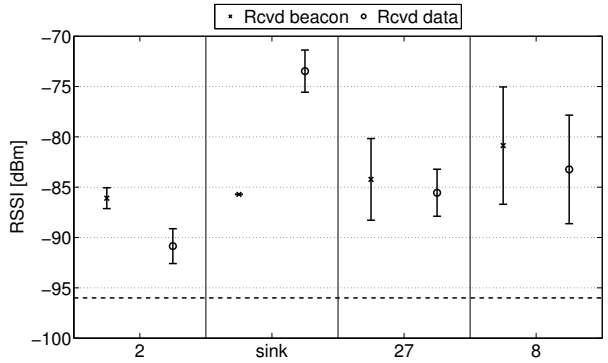


Figure 10. RSSI measurements at Jungfraujoch from the view of node 7. During the observation period, node 7 has been connected to four different parents. While node 27 was chosen as parent in 70% of the time, nodes 2 and 8 were each chosen in 15% of the time. Only a few tens of packets were directly transferred between node 7 and the sink.

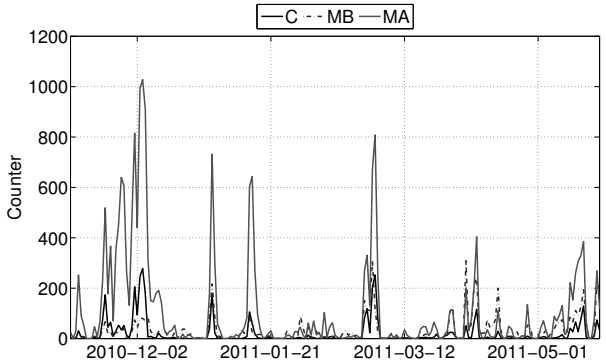


Figure 11. Event counters of node 7 at Jungfraujoch. Firstly, we can observe that packet acknowledgements (MA) are three times more often not received when compared to missed beacons (MB). This is an result of the uplink being significantly worse than the downlink when node 2 is chosen as the parent, see Figure 10.

B. Link Characteristics

The Semtech SX1211 radio used in TinyNode 184 nodes allows to measure the received signal strength (RSSI). The absolute measurement accuracy is specified at ±3dB [12]. Note that our radios operate with a static, maximum transmission power level. In our application, we can measure the RSSI in both directions of communication between two nodes. While the RSSI of the downlink (parent to child) is measured on the reception of a beacon message, the uplink (child to parent) RSSI on the reception of a data packet at the parent.

By adding network topology information, we firstly annotate each single RSSI measurements with the information if it was measured on the reception of a beacon message, or on the reception of a data packet. Annotated RSSI measurements are then grouped into links, where a link is described by the tuple *(s, r, t)* with sending node *s*, receiving node *r*, and type $t \in$ [`beacon, data`]. For each link, we calculate the mean RSSI $\bar{R}(s,r,t)$ and the standard deviation $\sigma_R(s,r,t)$. Note that we obviously only record results for successful packets and do not sample the surrounding noise floor. Our current system configuration requires a received signal strength of >-96dBm for packet reception.

The mean RSSI $\bar{R}(s,r,t)$ is then used to classify if a link is good, medium, or bad. Note that even medium links have a signal strength that is considerably larger than the receiver threshold. The resulting link characteristics based on RSSI measurements of three deployments from June 2010 to May 2011 are presented in Table II. Besides of the number of links for each class, we also show the RSSI variability by giving average and maximum standard deviation $\bar{\sigma}_R$ and $\forall(s,r,t) : \max \sigma_R = \max \sigma_R(s,r,t)$, respectively. While communication is usually bidirectional, there are also a few unidirectional downlinks that are an artifact of overheard beacon messages during parent search and normal operation.

As Table II shows most of the links are medium or good. Especially the Thur deployment features almost only good links. Due to the good quality of the links, the Thur deployment operates very stable. The table also indicates that there are medium and good links that have a very high variability. This variability may make links fail intermittently. Dozer is susceptible to such intermittent failures since two successively missed beacons or data uploads result in re-initiating a parent search.

Uplinks and downlinks have similar link distributions. For further analyzing the symmetry between uplinks and downlinks, we calculate the difference $d(s,r) = \bar{R}(s,r,\texttt{beacon}) - \bar{R}(r,s,\texttt{data})$ between averaged downlink and uplink measurements of bidirectional communication. The related results are presented in Table III. We see that there is a higher variability in asymmetry for the Jungfraujoch. In particular, this high

Table III
LINK SYMMETRY OF BIDIRECTIONAL COMMUNICATION BASED ON RSSI MEASUREMENTS FROM JUNE 2010 TO MAY 2011

Metric	$\bar{d}(s,r)$	$\sigma_{d(s,r)}$
Matterhorn	-0.0	1.7
Jungfraujoch	-0.1	2.9
Thur	0.3	1.3

variability is mainly due to a small number of links. For instance, the communication between nodes 2 and 7 is highly asymmetric with the beacon RSSI considerably larger than the data RSSI, see Figure 10. As a result, we can observe a considerably high amount of packet acknowledgements (MA) being not received in this constellation, see Figure 11. This can mean two things: packets are lost or the acknowledgments themselves are lost. Considering that the number of generated packet duplicates on this link being within the average range over the whole network, we have additional evidence that the parent is not receiving data packets properly. The question arises why this particular link is still chosen in 15% of the observation time instead of being marked as lossy. For once, there are four potential parents Dozer can choose from. A characteristic of the current implementation of Dozer is that it rules out potential parents with a RSSI below a certain threshold, but otherwise rates potential parents only based on their load and the distance to the sink. Since the beacon RSSI can be considerably better than the data RSSI, as seen with the aforementioned asymmetric links, this policy can result in suboptimal behavior. Finding and evaluating better rating functions is a topic for future work.

C. Data Yield

Table IV shows the results of our yield analysis. The data yield for data being generated between June 2010 and May 2011 is $\geq$99.53% for all three deployments, almost no packets are lost from the Thur deployment.

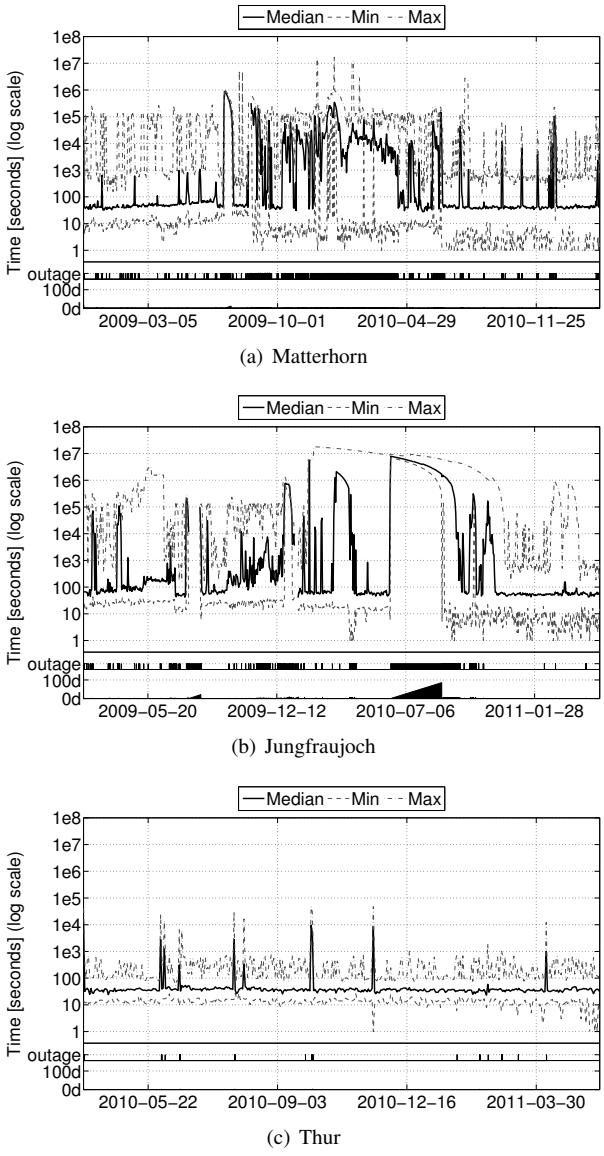
While the performance of single sensor nodes is almost equal over the whole network in the case of Matterhorn and Thur, we can see that the network performance at Jungfraujoch is dominated by a few sensor nodes that are suffering from bad connectivity. As our system model of the Dozer protocol is currently not fully able to explain the relation between bad connectivity and packet loss, we will have to further investigate this situation.

D. Packet duplicates

Concerning packet duplicates, our system model expects duplicates being generated if packets are (falsely) retransmitted by a child due to a lost acknowledgement from the parent. Under this assumption, the ratio between received unique packets and removed duplicates clearly proves Jungfraujoch and Matterhorn being more challenging environments when compared to the Thur deployment.

E. Duty cycle

The maximum duty cycle, which is related to the network lifetime, and the median of the distribution of the duty cycle



(a) Matterhorn

(b) Jungfraujoch

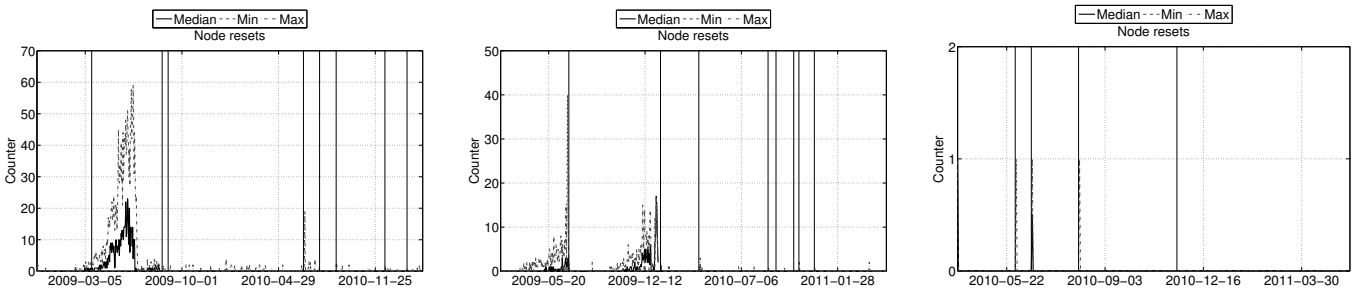
(c) Thur

Figure 12. Elapsed time on arrival at the base station and base station outages. Base station outages are shown at two scales: First, the upper bar contains entries for any outage with a duration of at least five minutes. Additionally, the second bar highlights outages with a duration of at least one day.

of single nodes are also presented in Table IV. Every two minutes, sensor nodes generate 2 to 8 packets of 32 bytes packet length, the data rate is 100 kbit/s. Regarding the median, we can clearly observe that the effort for achieving the seen data yield $\geq$99.00% is significantly higher at Matterhorn and Jungfraujoch deployments when compared to the Thur deployment. Found maximum duty cycles can possibly be explained by varying packet forwarding workload and single nodes suffering from bad communication links.

F. Packet Delay

Ideally, packets only spend a few seconds in the multi-hop network before they arrive at the sink. There are several causes of high packet delays: 1) Packet delays increase in the



(a) Matterhorn (b) Jungfraujoch (c) Thur

Figure 13. Node resets per sensor node and day. Deployment activity with at least one sensor node being replaced is denoted by vertical lines.

Table IV
PERFORMANCE STATISTICS FOR JUNE 2010 TO MAY 2011

Deployment	Matterhorn	Jungfraujoch	Thur
Packet counters, mean per node			
Unique	1,117,659	1,117,338	915,903
Missing	667	5,368	29
Duplicates	59,333	85,901	11,293
Data yield			
∅	99.94%	99.53%	100.00%
min	99.88%	96.28%	99.99%
max	100.00%	100.00%	100.00%
Radio duty cycle			
max	0.26%	1.31%	0.12%
Median	0.14%	0.34%	0.07%

case of local, high network contention when high amounts of forwarding traffic are routed through a small number of sensor nodes. 2) Harsh environmental conditions, *i.e.*, ice and snow, can lead to single sensor nodes not being able to communicate for several months. 3) Since communication is always initiated by the sink, outages of the sink lead to all sensor nodes not being able to communicate until this incident has been recovered. 4) Bad connectivity can lead to sensor nodes having lossy links resulting in more connection setup and packet retransmission efforts. The resulting lower throughput in terms of packets over time increases packet delays.

For analyzing packet delays, we calculate the maximum, minimum and average packet delay of all packets that a node generated during one day. Figure 12(a)-(c) show the absolute minimum and maximum packet delays of all sensor nodes. Additionally, the median of the averaged delays of all nodes is given. Here, using the median turned out to be more robust towards single nodes experiencing very large packet delays.

Figure 12(a)-(c) also show base station outages at two scales. While all incidents are shown in the upper bars, the lower bars highlight cases in which a base station was not working for multiple days. Combining base station outage and packet delay information confirms the expected causality. Data generated on all sensor nodes is experiencing large packet delays in case of a base station outage.

In Figure 12(b), a 82 days lasting base station outage during summer 2010 gives us a detailed insight into the buffering behavior of PermaDozer. On the start of the outage on the 11th of June, 2010, the median of the packet delay jumps to to 10^7 seconds (115.7 days). It is linearly decreasing until the recovery on the 1st of September, 2010. Afterwards, the median decreases almost exponentially until it is again down to 100 seconds. Further variations of the curve are then caused by consecutively following, shorter outages of the base station. It is now interesting to see, that packets were buffered up to 115.7 days and thus considerable longer than the duration of the outage. While the time measurement error due to drifting clocks is of course considerable higher for such long time spans, this observation also clearly shows us the delay of the protocol during flushing.

The base station at the Thur deployment being most stable, several short base station outages in the range of some hours can be easily detected by the peaks found in the measured packet delays, see Figure 12(c).

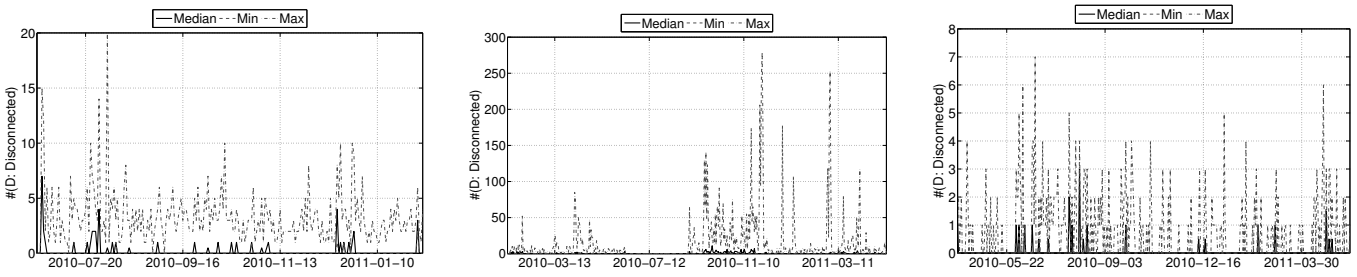
G. Node Resets

The number of node resets per sensor node and day is shown in Figure 13. The rapidly increasing numbers of resets visible between June 2009 and August 2009 in Figure 13(a), and between September 2009 and January 2010 in Figure 13(b), respectively, are related to a software bug that we reported in [8]. After deploying new nodes with attached SD cards for external storage, a bug in the filesystem driver produced a correlation between the amount of stored data and the execution time of write operations. Resulting task queue overflows triggered many watchdogs resets with growing occurrences over time.

As a result of our efforts in improving sensor node software and hardware, we can now see human intervention as the only cause of node resets. Concretely, all observed reset events at the Thur deployment can be explained by deployment activities that are marked by vertical bars in Figure 13(c).

VI. RELATED WORK

In [15], Zhao *et al.* study the packet delivery performance of a dense wireless sensor network in three different environments. Using 433 MHz radios, this work strictly focuses on packet reception and loss. The network performance over several months of system operation is evaluated in various deployment papers [14], [1], [5]. A study that focuses on the quality of wireless links in a challenging outdoor environment is presented in [4]. In [11], Mottola *et al.* study the impact



(a) Matterhorn (b) Jungfraujoch (c) Thur

Figure 14. Minimum, maximum and average number of disconnects (D) per node and day. The maximum values differ by up to two orders of magnitude.

of multi-path effects and vehicular traffic by comparing the performance of wireless sensor networks that are deployed in road tunnels.

VII. Conclusions

In this paper, we presented a comparative study on the performance of the PermaDozer application in three diverse deployments. Concerning reliability and data yield, very good results are achieved with the recent hardware and software generations that are already in use for more than one year.

However, it becomes also clear that the system performance is highly affected by the environment in which a sensor network is situated. We do not only have to deal with weak signals during packet reception, but also with asymmetric links affecting the system performance. Depending on the characteristics of a given environment, the effort for achieving a data yield $\geq 99.5\%$ can significantly vary. For example, the number of reported disconnect (D) events in Figure 14 is an order of magnitude higher at Jungfraujoch when compared to Matterhorn and Thur networks. Not only affecting the number topology changes, our performance analysis has also shown varying effort in terms of packet retransmissions, a varying number of packet duplications, and thus an also varying, resulting duty cycle.

Acknowledgements

We want to thank the anonymous reviewers for their valuable feedback that helped us to improve this paper. We also want to thank Stephan Gruber for his help during the preparation of the final version of this paper. The work presented was supported by NCCR-MICS, a center supported by the Swiss National Science Foundation, under grant number 5005-67322, and CCES RECORD, Philipp Schneider, for the Thur deployment.

References

[1] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments." in *Proc. 6th ACM Conf. Embedded Networked Sensor Systems (SenSys '08)*, 2008, pp. 43–56.

[2] J. Beutel, S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, I. Talzi, L. Thiele, C. Tschudin, M. Woehrle, and M. Yuecel, "PermaDAQ: A scientific instrument for precision sensing and data recovery in environmental extremes," in *Proc. 7th Int'l Conf. Information Processing Sensor Networks (IPSN '09)*, 2009, pp. 265–276.

[3] N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: ultra-low power data gathering in sensor networks," in *Proc. 6th Int'l Conf. Information Processing Sensor Networks (IPSN '07)*, 2007, pp. 450–459.

[4] M. Ceriotti, M. Chini, A. L. Murphy, G. P. Picco, F. Cagnacci, and B. Tolhurst, "Motes in the jungle: lessons learned from a short-term wsn deployment in the ecuador cloud forest," in *Proc. 4th Int'l Conf. on Real-world Wireless Sensor Networks (REALWSN '10)*, 2010, pp. 25–36.

[5] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment," in *Proc. 8th Int'l Conf. Information Processing Sensor Networks (IPSN '09)*, 2009, pp. 277–288.

[6] H. Dubois-Ferrière, L. Fabre, R. Meier, and P. Metrailler, "Tinynode: a comprehensive platform for wireless sensor network applications," in *Proc. 5th Int'l Conf. Information Processing Sensor Networks (IPSN '06)*, 2006, pp. 358–365.

[7] A. Hasler, I. Talzi, J. Beutel, C. Tschudin, and S. Gruber, "Wireless sensor networks in permafrost research - concept, requirements, implementation and challenges," in *Proc. 9th Int'l Conf. on Permafrost (NICOP 2008)*, vol. 1, 2008, pp. 669–674.

[8] M. Keller, J. Beutel, A. Meier, R. Lim, and L. Thiele, "Learning from sensor network data," in *Proc. 7th ACM Conf. Embedded Networked Sensor Systems (SenSys '09)*, 2009, pp. 383–384.

[9] M. Keller, L. Thiele, and J. Beutel, "Reconstruction of the correct temporal order of sensor network data," in *Proc. of the 10th Int'l Conference on Information Processing in Sensor Networks (IPSN '11)*, 2011, pp. 282–293.

[10] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, 2005, ch. TinyOS: An Operating System for Sensor Networks, pp. 115–148.

[11] L. Mottola, G. P. Picco, M. Ceriotti, c. Gună, and A. L. Murphy, "Not all wireless sensor networks are created equal: A comparative study on tunnels," *ACM Trans. Sen. Netw.*, vol. 7, pp. 15:1–15:33, September 2010.

[12] Semtech SX1211 Datasheet, http://www.semtech.com/images/datasheet/sx1211.pdf.

[13] TinyNode 184 Datasheet, http://www.tinynode.com/?q=system/files/SH-TN184-103_rev1.1.pdf.

[14] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, "A macroscope in the redwoods," in *Proc. 3rd ACM Conf. Embedded Networked Sensor Systems (SenSys '05)*, New York, 2005, pp. 51–63.

[15] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. 1st ACM Conf. Embedded Networked Sensor Systems (SenSys '03)*, 2003, pp. 1–13.