RZ 3596 (# 99606) 06/06/05
Electrical Engineering 28 pages

Research Report

On Architectures of Transmit-Only, UWB-Based Wireless Sensor Networks

Božidar Radunović, Hong Linh Truong,* and Martin Weisenhorn

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

*Email: hlt@zurich.ibm.com

Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

On Architectures of Transmit-Only, UWB-Based Wireless Sensor Networks

Božidar Radunović, Hong Linh Truong, Martin Weisenhorn
IBM Zurich Research Laboratory

March 15, 2005

Abstract

We are interested in designing a pulse-based Ultra-Wide-Band (UWB) sensor network. This network consists of a large number (in the order of 100) of *wireless sensor nodes (SNs)* that sense the environment and transmit the resulting data, and several (1-10) *cluster heads (CHs)*, that collect the data packets sent by the SNs and forward them to a *central server (CS)* for further processing. The goal is to have a network with simple and low-cost SNs that can support low data traffic rate (10 kbps on average). An UWB wireless receiver circuit in this case is much more complex than a transmitter one, and it is prohibitively expensive to integrate a receiver in a simple SN. We thus focus on sensor networks in which the SNs can only transmit data to CHs (transmit-only SNs), and we are interested in the optimal architectures of the SNs, CHs and CS such that the number of packets sent by the SNs and received successfully by the CS is maximized.

At the physical layer we decide to use a non-coherent receiver with an energy detector because of its implementation simplicity and low-power consumption. Furthermore we prefer a robust modulation scheme over those with high data rates and select 2-PPM as modulation scheme. At the MAC layer we propose a novel power-aware multi-access scheme that allows both the SNs and the CHs to turn off their radio transceivers and save energy during idle periods.

We find that the system performance can be drastically increased by introducing a detection threshold at the CHs: only packets whose received power is larger than a certain detection threshold are to be captured. By using an adaptive scheme that varies the detection threshold proportionally to the total traffic load generated by the SNs, we show that the system performance can be doubled during high traffic bursts without additional cost at the receiver. We also show that an additional improvement can be made by introducing an extra detection circuit, which detects packets with stronger power and switches the main receiver to that packet when it happens. We also find that combining data received from several CHs at the central server improves the coverage range without decreasing the throughput. Finally, we find that FEC coding of packets does not improve performance.

1 Introduction

1.1 Wireless Sensor Networks

There is a recent increase in interest for wireless sensor networks, due to its simplicity, low cost and easy deployment. Those networks can serve for different purposes, from measurement and detection, to automation and process control.

A typical wireless sensor network consists of a large number of sensor nodes (SNs) and a few sinks. SNs are wireless nodes equipped with sensing devices whose goal is to gather data and transmit it to a central server (CS) where the gathered data is processed. It is important that the transmission is wireless, since the number of sensors is typically very large, and the cost of deployment of a wired infrastructure is prohibitively expensive.

In order to have a long life time, SNs typically use small transmission powers. The area covered by a sensor network may be large, hence we need intermediate devices to relay data. These devices are called cluster heads (CHs). A CH is a device whose task is to capture transmissions of SNs in its environment, optionally do some limiting processing of the data, and forward it to a central server. One CH is responsible for coordinating a number of SNs: for a network of 100 SNs, we envisage to have less than 10 CHs, depending on the network area. Since there are much fewer CHs than SNs, they can be more expensive. They can rely on more sophisticated wireless technology to transmit data to the central server, or in some cases they can also be wired. In this work we focus only on the communication between the SNs and CHs, and we assume that all CHs have reliable (wired) links to the CS.

1.2 Ultra-Wide-Band Physical Layer and Coding

One of the promising physical layer technologies for future wireless sensor networks is the ultra-wide-band (UWB) physical layer. The characteristic of UWB is that it uses a large bandwidth, typically of order of several GHz, which allows to transfer data at high data rates while using low transmit power levels. Even though sensor applications typically do not require very high data rates, the whole network may require a high aggregate data rate due to a large number of simultaneously transmitting SNs.

One particular implementation of UWB is a pulsed-based UWB physical layer. It consists of sending very short pulses (of order of 1ns). A benefit of this technology, derived from radar systems, is an accurate distance estimation. Sensor networks based on pulse-based UWB are location aware, which is an important feature for applications like location tracking and intrusion detection.

Another benefit of pulse-based UWB architecture is a simple transmitter architecture. A typical modulation scheme for such physical layer is 2-PPM. A transmitter needs a pulse generation circuit, and the position of a pulse is a simple function of a transmitted symbol. On the contrary, an alternative UWB technology based on OFDM requires a much more complex transmitter that will generate multiple carrier frequency and distribute the load accordingly.

A pulse-based UWB receiver is a significantly more complex circuit. There are two main types of receivers: coherent and non-coherent. A coherent receiver achieves high data rates, but it needs to estimate the channel impulse response and a very accurate synchronization. On the contrary, non-coherent receiver does not estimate channel and needs less accurate synchronization. It has a simpler architecture, but it yields lower data rates. We assume that SNs and CHs are equipped with a non-coherent UWB physical layer that is described in [7].

Fundamental design parameters of a physical layer are transmitting power, coding and rate. In order to achieve long range communication, one has to use high transmission power or powerful codes to cope with signal attenuation. However, due to regulatory limit, high transmission power implies longer delays between pulses and thus a lower data rate. The same holds for coding: more powerful codes are more error-prone but decrease the rate of communication. Our choice of these parameters are explained in detail in Section 2.

1.3 Transmit-Only Sensor Nodes

A sensor network comprises a large number of SNs. It is thus important that these nodes are as simple and as low-cost as possible. We want a sensor network to support relatively high data rates and location capabilities, and focus on a pulse-based UWB physical layer. As discussed in Section 1.2, pulse-based UWB transmitters are low-cost and simple to implement. Nevertheless, even a simpler, non-coherent receiver, requires complex elements, such as synchronization circuit, and may be prohibitively expensive for low-cost SNs.

Therefore, we assume a network of transmit-only SNs, equipped with sensing and transmitting devices. These SNs measure some data and transmit it to CHs. The SNs cannot sense the medium nor can they receive any feedback from CHs or other SNs, hence SNs are completely unaware of the global state of the network. This choice of SN architecture implies that most of the design complexity is in the CH and the CS.

1.4 System Requirements

Sensor networks are usually low data rate networks, as described in [6]. The main reason is that low traffic, hence low *average* data rates imply low power dissipation and long network lifetime. However, we emphasize that we are talking about low average data rate. The peak traffic may still be high, but only during very infrequent time intervals. A typical sensor traffic thus may vary from a few packets per hour up to 400 kbps for video transmissions. Note that these numbers represent average data rates: sensors will transmit packets at physical layer data rate (which is of order of MBps), and the average rate will depend on time gaps between packets.

A typical network consists of up to hundreds of sensors. Therefore, even if a video transmission from a single sensor is considered low traffic, a simultaneous video transmissions of tens of sensors is several times larger than the rate of the physical layer itself. A network should thus be designed in such a way that it can maximize its performance both during low traffic intervals and high traffic bursts.

We assume there exist low- and high-priority SNs. High-priority SNs are located near the CHs and are expected with high probability to successfully transmit packets. Low-priority SNs are expected to deliver packets only when the total traffic load is low and may be placed far away from the CHs. This facilitates the deployment of a network and makes it more cost effective.

1.5 Application Scenarios

In order to better understand the system requirements, we illustrate them on by using an example of a surveillance system, based on scenario 21 from [6]. An underground car park is filled with SNs. There are several types of SNs. Some are of low priority, like those for temperature and humidity measurements. They generate very low traffic (< 10 kbps) and one or a few transmissions can be lost. Other SNs are of high priority, like seismic, infrared and microphone SNs that are used to detect movements of an intruder (≈ 10kbps traffic), and cameras that are transmitting live videos from the area (≈ 400kbps traffic). Typical network of this type consists of 10-100 SNs, and when cameras are active the aggregate rate may go up to several tens of Mbps. The scenario is depicted in Figure 1.

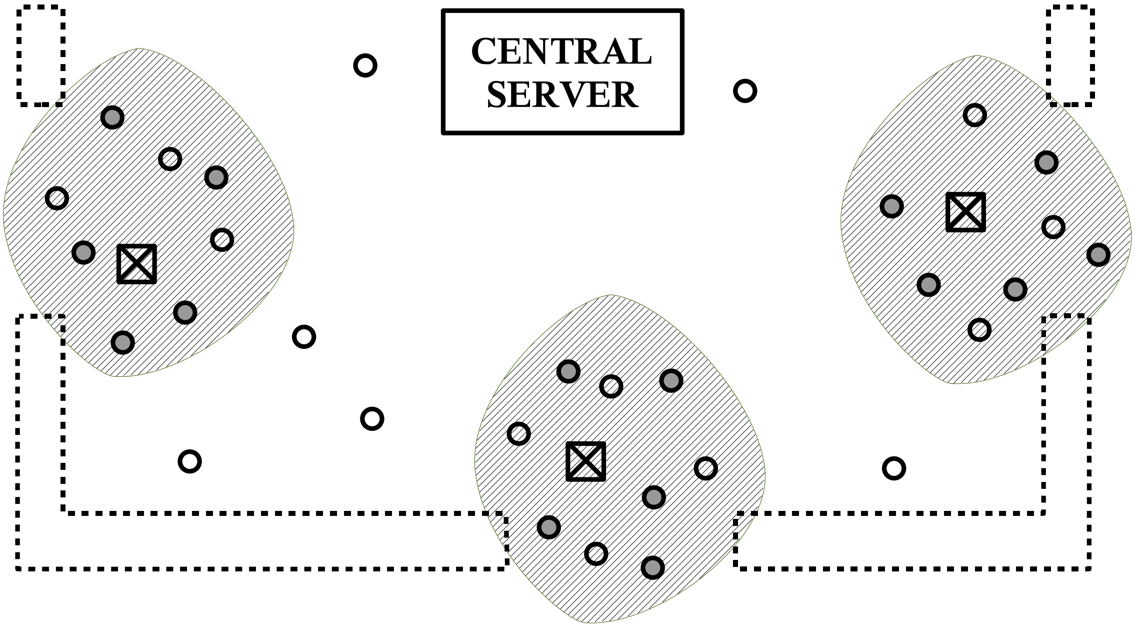

Figure 1: An illustration of a transmit-only sensor network in the intrusion detection scenario. SNs are denoted with circles, and CHs with crossed boxes. Empty circles are low-traffic SNs and solid circles are video SNs. SNs that are placed in shaded areas around CHs are high-priority SNs. Others are low-priority SNs.

Camera and movement detection nodes are placed near the CHs. When the traffic is high, there will be a lot of collisions between SNs' transmissions. Since SNs are unaware of the current network traffic intensity, collisions cannot be prevented. However, if high-priority SNs are close to the CHs, interference from distant transmissions is going to be low compared to the received signal power, hence the packet error rates are going to be low. On the contrary, low-priority SNs may be significantly farther away. Their packets will be correctly received only when there is no intrusion detection, which is sufficient for this type of application.

Similar frameworks are described in scenarios 15 and 26 of [6]. Scenario 15 discussed position monitoring for training purposes. A typical network contains 100 SNs and the maximal rate is 100 kbps. Scenario 26 presents a smart shelf management and monitoring system. The system is required to accommodate up to 1000 nodes with rates of 10-100 kbps. Although hardly ever all nodes will be active at the same time, the total aggregated input traffic of the network can easily go to tens of Mbps.

A similar example is a fire detection sensor network [5]. SNs are distributed on an area, and their goal is to detect a fire, and to monitor its spreading. Normally, the traffic in such a network is very low. However, in case of fire, there is a burst of packets transmitted by those SNs that detect the fire. Most of these information are redundant to some extent: it is sufficient to get packets from one SN to detect the fire. In order to get more precise information on fire spreading, we need to capture more packets.

Another important application parameter is the communication range. As described in [6], a range of communication in LOS for this type of applications is from 10m to 100m. We select the target communication range to be 60m. Network coverage can be further improved by deploying more CHs.

1.6 Performance Metrics

Summarizing the above requirements, we focus on sensor networks with low average data rates and a large number of SNs, but with high peak data rates. Our goal is to develop network architecture that will be available to sustain the bursts periods with peak transmission rates, defined by these examples, and which will at the same time be efficient during low-traffic periods.

When the traffic is low, collision probability is low. In this case the goal of the CHs is to capture packets from as many SNs as possible, thus to cover the largest possible area. Therefore, in low-traffic regime, our performance metric is **range maximization**.

On the contrary, when the traffic is high, there will be a lot of collisions. All SNs are transmit-only, hence they cannot sense the actual traffic intensity and avoid collisions. In this regime, CHs should concentrate only on receiving data from high-priority SNs in their neighborhood and maximize the total number of packets they can capture from these SNs. Thus, in high-traffic regime, our performance metric is **throughput maximization**.

The throughput maximization metric does not explicitly consider fairness issues. By maximizing throughput some distant SNs may starve. However, some form fairness is already implied by the network topology design itself. As described in application requirements, high-priority SNs are expected to be placed near to the CHs. Therefore, all high-priority SNs will get approximately the same attention, while low priority SNs will only starve during traffic bursts (which is one of the design assumptions). More discussion on performance metrics can be found in Section 5.1.

1.7 Sensor Node (SN) Architecture

As explained above, since SNs are transmit-only devices, their MAC layer is extremely simple. SNs do not know the state of the network so their medium access is based on local decision. We propose several scheduling strategies to improve the capacity and decrease CH's power dissipations.

Another important aspect of SN architecture is to choose an appropriate coding and signal power. The average UWB signal power is limited by regulations. If one wants to increase the rate, i.e. send more pulses per second, than the transmit energy of a pulse has to be decreased.

This in turns yields lower communication range since distant CHs will not be able to detect weak pulses.

In order to receive a packet, a CH first needs to synchronize to it. This is possible if the bit error rate is lower than 10^{-1} [7]. Once synchronized, a packet is correctly received if there are no bit errors. A forward error correction (FEC) code can be implemented to protect payload from errors and to increase sustainable bit error rate. Another way to address the range/rate trade-off is to change coding. More powerful code will also increase range but will decrease rate.

Issues arising in SN architecture are thoroughly explained in Section 2.

1.8 Cluster Head (CH) Architecture

Once a packet is transmitted from a SN, it will be successfully received by a CH if the signal strength is high enough, and if the level of interference coming from concurrent transmissions is low enough.

The goal of the CHs is to successfully receive as many packets as possible. If the sensor network is lightly loaded, the optimal strategy of a CH is trivial: it should try to receive every packet it can detect. Since CHs do not control the medium access of the SNs, they cannot prevent transmission failures that occur due to collisions.

However, the story is different when the network load is high. A typical wireless receiver has only a single receiving circuit, thus can receive only one packet at a time. While a CH is receiving a packet from a distant SN, another transmission may start from a near-by SN. This new transmission will interfere and may corrupt the packet being received. At the same time, the CH will not be able to receive the interfering, new packet since its receiving circuit was busy when its transmission started. Hence, both packets will be lost.

This problem can be overcome if a CH is equipped with several receiving circuits. However, such a solution is expensive and difficult to implement. In Section 3 we present alternative CH architectures that alleviate this problem.

1.9 Central Server (CS) Architecture

All CHs send data they received from the SNs to the CS. In the simplest approach each CH tries to decode a received packet. If the decoding succeeds, it transfers the decoded packet to the CS. Otherwise, it discard the received information.

However, it is known from the theory of multi-antenna systems that different ways of combining can improve the packet reception. In particular, multiple CHs attached to a CS can be viewed as a multiple input antenna system. Each CH thus does not decode a packet, but sends demodulated soft samples to the CS. The CS combines the received samples and then performs the decoding. We consider both a simple architecture with no combining, and one with a maximum ratio combining principle which is know to be optimal in multiple-antenna systems with Gaussian noise. We explain this issue in more details in Section 4.

1.10 Problem Definition and Main Findings

We consider a pulse-based UWB sensor network with transmit-only SNs, and we seek for networking architectures that will maximize the number of captured packets.

We first consider a choice of the optimal transmit power and coding at SNs. We select minimum transmit power to still be able to synchronize at the required communication range, defined in Section 1.5. We then numerically evaluate the performance of different coding schemes. In most of the cases it is optimal to use no additional coding. However, for certain CH architectures and high loads, it is optimal to use coding.

We describe various multiple access schemes that are applicable to our transmit-only SN architecture. In particular we propose a novel power-aware multi-access scheme that allows both the SNs and the CHs to turn off their radio transceivers and save energy during idle periods.

Next, we study the performance of three different CH architectures. The first is a conventional one where a CH tries to receive any packet to which it manages to synchronize. We call this *CH architecture with no threshold*.

The second one is based on an adaptive detection threshold. Each CH has a single receiver circuit with a detection threshold which is adapted as a function of the traffic sent by the SNs. It starts receiving a packet only if the received signal power is above the detection threshold. In addition, the CH tracks the incoming traffic intensity and constantly adapts the detection threshold to the actual traffic load. We call this *CH architecture with adaptive threshold*.

The third architecture assumes that each CH contains one receiving circuit, and an additional detection and synchronization circuit. A CH starts receiving the first packet it observes on the wireless medium. The goal of the additional detection circuit is to monitor the medium in parallel, and to detect if a packet, stronger than the one currently being received, appears. If this happens, the receiving circuit drops the ongoing transmission and switches to the stronger packet. We call this *switched CH architecture*.

We analyze the performance of the proposed CH architectures in conjunction with different codes and CS architectures. When total traffic is low, we find that combining at the CS can increase the range for up to 20m. When traffic is high, we find that the approach with the adaptive detection threshold yields great improvements comparing to the simple architecture, while maintaining the same level of architectural complexity. The switched architecture introduces an additional performance improvement but with a slight increase in CH production cost.

2 Sensor Node (SN) Architecture

There are four parameters that define the SN architecture: packet sizes, transmit power, coding, and medium access. They are described in the following.

2.1 Packet Sizes

SNs typically send small chunks of data. Here we assume packet size is fixed to 100 bytes (800 bits) with preambles. Similar performance results would be obtained with different packet sizes.

2.2 Transmit Power

As defined in Section 1.5, we require communication range to be 60m. The upper limit on power spectral density of a UWB signal, defined by FCC, is -41.25 dBm/MHz. The goal is to transmit data at a power sufficiently high such that a CH 60 meters away can synchronize to the signal. As described in [7], synchronization is possible if the bit error rate is lower than 10^{-1}.

We choose pulse energy to be 30 pJ, which implies that the time between consecutive pulses is 400 ns. This in turn yields a rate of 2.5 Mbps. The bit error rate at 60m links is around 10^{-1}. Although this is sufficient for synchronization, it is not enough for successful packet receptions. Additional forward error correction or signal combining has to be performed to cope with this error rate. This is described in the following subsection and in Section 4.

2.3 Coding

As mentioned before, the transmit power is tuned to achieve a target bit error rate of 10^{-1} at 60m links. In order to receive a packet at that distance, it is necessary to use some form of forward error correction.

We use a simple model of coding. We assume the underlying channel between a SN and a CH is a binary symmetric channel [2]. This means that every bit at the output will be flipped with some probability p, called error probability. The capacity of this channel is

$$C(p) = 1 - H(p) = 1 + p\log_2(p) + (1-p)\log_2(1-p).$$

In other words, we can construct a code of infinite block length that will be able to achieve capacity $C(p)$. For example, if error probability $p = 0.1$, the achievable capacity is $C(p) = 0.5$ which means we can transmit 0.5 bits per channel use, or in order to convey n bits of information, we need to transmit $2n$ symbols. We also say that the code rate in this case is 1/2.

We assume that during a design time we can select a maximum sustainable error rate p. We then construct a code to cope with that error rate. The end-to-end data rate will be the physical data rate multiplied by $C(p)$. Again, if we want to cope with 10% error rate, the end-to-end bit rate will be 1.25 Mbps (the physical data rate is still 2.5 Mbps but in order to transmit a packet of 800 bits we need to send 1600 coded symbols).

Since the maximum tolerable error rate for synchronization is 10%, there is no need to consider codes for higher error rates than that. In the performance analysis part we will evaluate performances of different codes in conjunction with different CH and CS architectures.

Note that our model of coding is just a simple approximation. A real implementation would apply coding on soft samples, and not on hard ones, as we assume here. This would increase the performance of codes hence possibly change some of our conclusion. An implementation of coding remains as a future work.

2.4 Multiple Access Schemes

While the physical layer defines how and when the pulses belonging to a certain PHY protocol data unit (PDU) are sent over the wireless medium, the multiple access scheme at the MAC layer governs how and when the SNs are allowed to access the medium, i.e. how and when they are allowed to start transmitting with the first pulse of their PHY PDUs.

Because of the lack of a receive capability at the SNs, all access schemes that require the SNs to sense the medium (e.g. CSMA) or to receive access control information (e.g. IEEE 802.15.4) cannot be applied. The SNs have to share the medium in an totally uncoordinated manner. Some possible access strategies are described in the following.

Note that all the access schemes described in the following subsections share the same ”weakness” of being able to operate efficiently only in a lightly loaded network; at high load they may lead to an useless system due to excessive collisions. However, it should be recalled that we are designing a LDR/LT sensor network, in which the main and critical design points are low cost, low complexity, low power consumption, and not high data rates.

Furthermore, since the SNs do not have a receive capability, the successful reception of a packet by a CH cannot be guaranteed. However, its probability can be increased by the use of appropriate channel coding (e.g. repetition or Reed-Solomon codes) and in particular by a sophisticated cooperation between the CHs. This will be discussed in Sections 4 and sec:perfeval.

2.4.1 Immediate Access (IA) Scheme

Whenever a SN has a packet ready to send, it just accesses the medium and sends it straight away. This simple strategy may work well in lightly loaded networks, has no access delay, but may lead to catastrophic collisions when multiple SNs have packets to be sent almost at the same time.

2.4.2 Random Access (RA) Scheme

To reduce the probability of catastrophic collisions when multiple SNs have packets to be sent almost at the same time, the SNs wait for a random time before they access the medium and transmit.

This scheme is similar to the well-known unslotted ALOHA method. However, due to the transmit-only characteristic of the SNs, there are no retransmissions due to collisions.

2.4.3 Power-Saving Scheduled Access (SA) Scheme

In many sensor applications, the SNs are idle for long time if no sensing event happens. To save energy the SNs can turn off their radio and sleep during those idle times. They need only to wake up when they have something to send. Although the IA or RA scheme described above is well appropriate for those low traffic conditions, it has the disadvantage of requiring the CHs to have their radio receivers always turned on, because they do not know when a SN would start sending, thus wasting energy for idle listening.

The scheduled access (SA) scheme described in the following will allow the CHs to switch their receiver off and save energy during those idle phases.

In its most general form, the SA scheme allows an SN to access the wireless medium only at scheduled time instants. When there is a packet to be sent, it is sent at the next scheduled time instant. If the SNs now inform the CHs about their next scheduled transmit instants, then the CHs could sleep during the idle periods and only wake up at those instants to receive the packets sent by the SNs. The SNs can inform the CHs about their transmit schedules by using one of the following methods:

- **Constant transmit intervals**

  All SNs transmit at constant and specific intervals and indicate in their data packets the remaining time until their next transmit instants. A CH first collects this information during its initialization phase. Then, based on this information it can go to sleep and wakes up at the scheduled transmit instants to receive data. If at a scheduled transmit instant a SN does not have any data to send, it just skips this instant. Since the intervals between two transmit instants are constant, the CHs can determine the next scheduled instant.

  To avoid catastrophic collisions the SNs can freely select their own transmit intervals, e.g. using RDMA [8]. In this case, they can indicate in their data packet the rate that they are using instead of the remaining time until their next scheduled transmit instant.

- **Pseudo-random transmit intervals**

  All SNs generate the time intervals between two consecutive transmit instants using a common pseudo-random generator with the same seed. Thus the sequence of the randomly generated transmit intervals are the same for all SNs. Furthermore this sequence is known by the CHs. The data packets sent by the SNs contain the index of the next time interval to be used, thus allowing the CHs to determine their next scheduled transmit instants. If the CHs miss one transmission of a certain SN (because of transmit collisions, or packet errors, or because the SN does not have any data to sent), the CHs can still determine the next transmit time of that SN based on the indexes they received in former data packets from that SN.

  To avoid catastrophic collisions the first index to be used is selected randomly by the SN.

  A variation of this method is that the SNs use different random generators and/or different seeds. In these cases information about the selected generator/seed needs also be indicated to the CHs, so that they can reconstruct the transmit time sequences.

3 Cluster Head (CH) Architectures

3.1 CH Architecture Based on Detection Threshold

We first consider a CH with a single receiving circuit and a variable detection threshold, which we denote with P_{dt}. If the signal strength of a received packet is lower than P_{dt}, then the packet will be ignored. Otherwise, the CH will try to receive it. We say that a packet is **detected** if the received signal is stronger than P_{dt}. Only then a CH will try to receive it.

For simplicity of presentation, one can assume that the signal attenuation is a time invariant function of distance and that all SNs send with the same power. Than there exists a threshold region of radius R_{dt}, such that if a SN is outside of this region, a CH will not start receiving packets sent by this SN. This is illustrated in Figure 2.

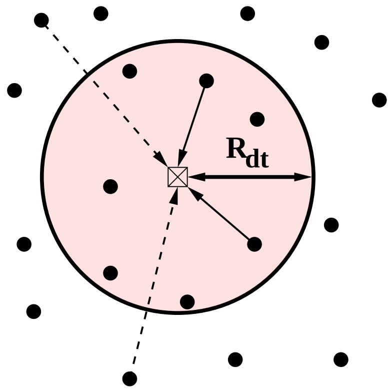

Figure 2: An illustration of the detection threshold. Transmissions of SNs whose received signal at the CH is higher than P_{dt} are denoted with solid lines. The transmissions of SNs whose received signal is below P_{dt} are denoted with dashed lines. The equivalent detection region of radius R_{dt} is represented with a shaded circle.

The main reason why a CH does not want to start receiving a packet is that if it starts receiving a packet whose signal strength is low, there is a high probability that the packet will be dropped due to collision. In the meantime, other packets, possibly with a higher signal strength, will be rejected since the only receiving circuit is busy.

An obvious drawback of this approach is that distant SNs will not be able to convey any information at all to the CS. However, this drawback is a consequence of the constraints on the SN architecture, and not of the protocol. Since SNs cannot adapt their medium-access policy, there is no way to receive packets from distant SNs when the traffic is high, regardless of the CH architecture. Nevertheless, this problem can be mitigated in several ways. Firstly, as one can see from the examples in Section 1.5, a burst of traffic is usually triggered by an event. Therefore, even if some packets from distant SNs are dropped, we might not loose too much information due to a correlation among data. However, if a reliability of information is crucial, a simple solution is to add more CHs on critical places, to maximize the capture probability. Since the number of CHs is anyway expected to be significantly lower than the number of SNs, the solution will be cheaper than implementing a receiver in each SN.

3.1.1 Theoretical Analysis

In order to better understand this issue, we define a simple model of the system and we analyze it analytically. First we assume that the traffic of every SN is Poisson. This is a somewhat reasonable assumption, since if a system is heavily loaded, the optimal medium access for every SN is to defer each transmission for some random time (similar to random backoff in ALOHA). We also assume a simplified physical layer model: if a CH receives a packet from a node at power P^{rcv}, and if an interfering packet, which overlaps even for a small fraction of time, comes with power larger than $P^{rcv} - \Delta$, then the received packet will be lost. We tested this approximation on physical model in [7], using networks with 2 nodes, and we found that that approximation holds. This simplification neglects the impacts of multiple concurrent interferences, but as we will see it fits well with the simulated results.

We now consider a scenario depicted on Figure 2 with one CH and n SNs. Signal from SN

i is received at the CH with power ${P^{rcv}}_i$. Each SN i generate a Poisson traffic with distribution λ_i. Let the detection threshold be P_{dt}, which means that we will try to decode packets whose received power is larger than P_{dt}. The total traffic generated by those nodes is

$$\lambda_a(P_{dt}) = \sum_{i:{P^{rcv}}_i \geq P_{dt}} \lambda_i.$$

We first estimate the probability that the receiver is idle at any given moment in time. The receiver is idle if it has finished decoding a previous packet (successfully or unsuccessfully), and if no other packet has arrived in the meantime with received power larger P_{dt}. The state of receiver (busy or idle) is a stationary process in time so we call the probability of receiver being idle $P_{\text{rcv idle}}$. We can describe this process with a continuous Markov chain and we get the stationary probability $P_{\text{rcv idle}} = 1/(1 + \lambda_a(P_{dt}))$.

We next model the probability that a SN i will successfully transmit a packet. This will happen if the receiver is idle at the time a packet transmission starts, and if the packet does not overlap with any packet whose received power is higher than ${P^{rcv}}_i - \Delta$. We assume all packets have a fixed length and we assume that the packet transmission time is one. Similar to non-slotted Aloha, we have that if a packet arrives at time 0, then any other packet arriving within $[-1, 1]$ will interfere with it and cause collision. Therefore, the packet capture probability of node i is

$$P_{\text{capture}}(i, P_{dt}) = P_{\text{rcv idle}}(P_{dt}) \exp(-2 \sum_{j:{P^{rcv}}_j \geq {P^{rcv}}_i - \Delta} \lambda_j).$$

The average throughput of SN i is then $\lambda_i P_{\text{capture}}(i, P_{dt})$ and the average throughput of all SNs is

$$\bar{X}(P_{dt}) = \sum_{i:{P^{rcv}}_i \geq P_{dt}} \lambda_i P_{\text{capture}}(i, P_{dt}) \tag{1}$$

The optimization problem of maximizing (1) can be solved numerically. We solved it for a large number of topologies and traffic distribution and we find that it is always optimal to maintain $\lambda_a = 0.75$ which yields the efficiency of the medium $\bar{X}$ of 25%. In other words, we should estimate $\lambda_a(P_{dt})$ and vary P_{dt} to obtain $\lambda_a = 0.75$.

We verify our model by simulations. We randomly distribute 50 SNs on 40m x 40m square and look at the goodput of the system for different load. The physical link rate is 5Mb/s, and we can see in Figure 3 that the goodput is maximal when the aggregate traffic is around 75% of the physical link rate. At that point, the utilization of the system is around 25%.

It may seems at the first sight that a simple model of non-slotted Aloha can be use to model the problem. However, in non-slotted Aloha, the maximum utilization is 18%, which is achieved when the total load is 50% of the physical fixed rate. These numbers have a high discrepancy with the simulation results from Figure 3, hence they cannot be used to design an efficient adaptive receiver.

3.1.2 Optimal Architecture of Adaptive Receiver

As described in Section 3.1.1, it is optimal to keep $\lambda_a = 0.75$. We propose a simple method to track load λ_a and utilization $\bar{X}$ of the system, and to adapt P_{dt} in order to keep utilization at the maximum. We give a simple example of packet arrivals in Figure 4 to illustrate the idea.

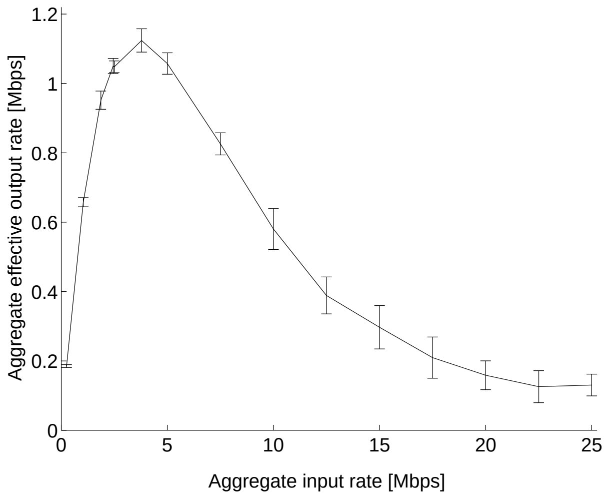

Figure 3: We consider 50 SNs uniformly distributed on 40m×40m square, with one CH in the middle. On x-axis we see the aggregate input SN traffic. On y-axis we see the aggregate goodput.

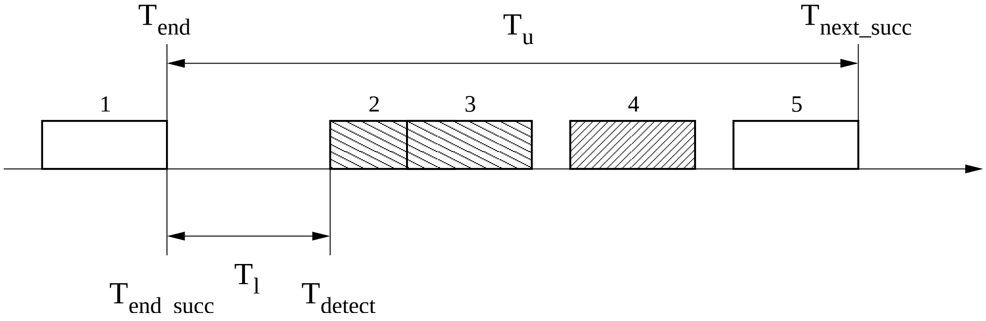

Figure 4: An example of packet arrivals that illustrates the adaptation mechanism. Packet 1 has arrived and is well received. Packet 2 was being received when it collided with packet 3 and is discarded. Packet 4 is not detected because it arrives from a node outside of a detection threshold. Finally, packet 5 again is well received.

We first show how to estimate $\lambda_a, \bar{X}$. As shown in Figure 4, we denote with T_l the average idle time of a receiver, that is the average time between two packets from sensors that are detected. We keep track of the end time T_{end} of the last detected packet (successfully or unsuccessfully received) and at the moment T_{detect} when we detect a new packet, we update $T_l = \alpha T_l + (1-\alpha)(T_{\text{detect}} - T_{\text{end}})$. The estimate of the intensity of detected load is then $\lambda_a = \text{packet\_duration}/T_l$.

Similar thing is done for estimating utilization $\bar{X}$. We denote with T_u the average time between two successful receptions. We then keep the time of the end of the last successful packet transmission $T_{\text{end\_succ}}$. At the instant of the next successful packet transmission $T_{\text{next\_succ}}$, we update T_u. For updating, we use exponential weighted average $T_u = \alpha T_u + (1-\alpha)(T_{\text{next\_succ}} - T_{\text{end\_succ}})$. The estimate of the utilization is then $\bar{X} = \text{packet\_duration}/T_u$. We set the filter constant $\alpha = 0.95$ in both cases.

In parallel, a CH also needs to learn about existing SNs and their distances. This is done during packet receptions. Note that a CH does not need necessarily to successfully receive a whole packet to perform this estimate. It might be sufficient to decode the header, and to estimate signal strength. As a result of this estimation a CH keeps a list of active sensors and their received powers $\{i, {P^{rcv}}_i\}_{i=1,\cdots,n}$, ordered decreasingly by powers.

The key idea of the algorithm is to keep utilization at 25% and detected load at 75%. In theory it should be sufficient to use detected load as an estimator, but we use both detected load and utilization to have better robustness. First we set $P_{dt} = 0$ and we are able to detect any SN. We start receiving packets, and we update T_l, T_u and the list of SNs. Initially, at the bootstrap, the estimated detected load and utilization are low. Once the detected load goes over 75% and at the same time the utilization drops below 25%, it means that we have passed over the top of the curve on Figure 4, hence P_{dt} is too small and has to be increased.

The decision on the size of the detection threshold happens every time a new packet is sensed on the medium. It is important to notice that it is more dangerous to overestimate the detection threshold than to underestimate it. This is due to the shape of the curve on Figure 3. If we overestimate the detection threshold when the total input traffic is low (left side of the curve), this means that we further decrease the input traffic hence further decrease the effective output. Similarly, when the total input traffic is high (right side of the curve), if we underestimate the detection threshold, we increase the total number of detected packets and again decrease the effective output rate. However, as we can see from Figure 3, the slope of the curve is much steeper for lower input rates, hence the potential loss when overestimating the detection threshold is higher.

Therefore, we perform a conservative decrease of P_{dt}: if it happens 4 times in a row that the detected load is higher then 75% and the utilization lower than 25%, only then we will increase the detection threshold by removing one SN from it (in other words, if we assume that $P_{dt} = {P^{rcv}}_i$, then we update $P_{dt} = {P^{rcv}}_{i-1}$, if $i > 1$).

On the contrary, when decreasing the detection threshold, we are less conservative. The first moment when the detected load is lower then 75% and the utilization is lower than 25%, we set $P_{dt} = {P^{rcv}}_{i+1}$ (if $i < n$, or else $P_{dt} = 0$).

Another important point is to keep updating T_l and T_u even when no packet arrivals are being detected. In case when a traffic intensity suddenly drops, or nearby nodes cease transmitting, we might have the detection threshold too high. If a new packet arrives at time T_{now}, we will take the following values of $\lambda_a, \bar{X}$:

$$\begin{aligned}\lambda_a &= \frac{\text{packet\_duration}}{\alpha T_l + (1-\alpha)(T_{\text{now}} - T_{\text{end}})},\\ \bar{X} &= \frac{\text{packet\_duration}}{\alpha T_u + (1-\alpha)(T_{\text{now}} - T_{\text{end\_succ}})}.\end{aligned}$$

This way, the detection threshold will gradually drop in time while there is no detected packet.

At the end, for completeness, we give the pseudo code of operations. Operation `start_transmission(j)` is called at a CH when a packet from node j is sensed. Operation `end_transmission(j)` is called when a packet transmission is finished. Note that `end_transmission(j)` is called only if a packet was detected (the received power is above the detection threshold P_{dt}).

start_transmission (from node j):

```
  lambda_a = packet_duration
           / (ALPHA * T_l + (1-ALPHA) * (now - T_end));
  X = packet_duration
    / (ALPHA * T_u + (1-ALPHA) * (now - T_end_succ));


  if (X < 0.2) count = count + 1;
  else count = 0;

  if (lambda_a < 75% and X < 20%)
  begin
    i = i+1;
    P_dt = Prcv_i;
    count = 0;
  end
  else if (count >= 4 and lambda_a > 75% and X > 20%)
  begin
    i = i-1;
    P_dt = Prcv_i;
    count = 0;
  end

  if (Prcv_j >= P_dt)
  begin

    // Receive

    T_l = ALPHA * T_l + (1-ALPHA) * (now - T_end);
    lambda_a = packet_duration / T_l;

  end
```

end_transmission:

```
  if (successfully_received)
  begin
      t_u = ALPHA * t_u + (1-ALPHA) * (now - last_rcv);
      util = packet_duration / utime;

      T_end_succ = now;
  end

  if (packet_was_detected) T_end = now;
```

3.2 Switched CH Architecture

As we have seen in the previous section, the main reason for low efficiency of a CH is that if it starts receiving a weak packet, and a stronger packet arrives during this reception, the weaker packet will be dropped due to the interference, and the stronger packet will not be received because the receiving circuit was busy when the packet arrived.

The most general way to solve this problem is to include several receiving circuits in parallel so that a CH can cope with all arriving packets. This is not necessary in most of the cases. We propose an alternative solution that offers a similar performance while being simpler and cheaper to implement.

The basic idea is to include another circuit for detection and synchronization, in addition to the full receiving circuit. This additional circuit is constantly monitoring the wireless medium for a newly arriving packets. If a transmission of new packet starts, if its signal is stronger than the signal of the packet currently being received, and if it significantly overlaps with the current packet, the CH stops receiving the current packet and switches to the new, stronger packet. We call this architecture *switched CH architecture*.

4 Central Server (CS) Architecture

The goal of a CS is to collect information from CHs about received packets. In its most simple implementation, a server only receives packets that are successfully decoded by at least one CH. If no CH successfully decoded a packet, the packet is lost.

In order to improve the performance of the system, we also propose a more advanced CS architecture. It is based on ideas from multi-antenna systems. Several CHs connected to a CS can be viewed as a multiple input antenna system. If a CH cannot decode the packet, it just send the demodulated soft samples to the CS.

If at least one CH successfully decodes the packet, there is no need for further processing. However, if all of them fail, the CS then combines the received samples from multiple CHs and tries decoding it.

It is known that the optimal combining for channels with additive white Gaussian noise is *maximum ratio combining* [1]. As we can see from the analysis in [7], our physical layer can be closely approximated with a 2-PAM channel with Gaussian noise, hence maximum ratio combining should also be the optimal combining. In this work, we will compare these two approaches: **single antenna approach** (the approach without combining) and the **maximum-ratio combining**.

In the presence of interference, the interference is no more Gaussian, hence the maximum ratio combining is not anymore the optimal combining. More advanced techniques, like minimum mean-square error (MMSE) receivers should be applied. However, this approach is difficult to pursue for two reasons. Firstly, it is difficult to derive the optimal receiver in case of interference, since the interference introduces the mixed terms (as explained in [7]), and is not purely Gaussian. Secondly, to design an optimal receiver we would need the perfect estimate of total interference at any point in time, which is difficult to implement.

Finally, as we will see in the performance evaluation section, combining is used to increase the range in case of low traffic. In that case, the dominant noise component is background

Gaussian noise. When the traffic is high, CHs will focus on nearby sensors, hence there will be no use in combining. For the above reasons we do not analyze more advance combining schemes but we focus solely on maximum ratio combining.

5 Performance Evaluation

In this section we numerically evaluate performances of the proposed concept using a home-made cross-layer simulator [4]. We first discuss in more details the performance metrics used in evaluation, and then we present results for each of the performance metrics.

5.1 Performance Metric

5.1.1 Range Maximization

As we have seen in the application requirements, described in Section 1.5, there are two main application scenarios for sensor networks. The first one is low-traffic scenario. Each SN sends packets sporadically and the probability of collision is very low. The goal of the CHs is then to capture packets from as many SNs as possible, thus to cover the largest possible area. Therefore, in low-traffic regime, our performance metric is **range maximization**.

We say that the **range of a network** is the maximum distance from the central point of a network at which a single SN sending packets will have more than 80% of captured packets. The central point of is defined as the circumcenter of a polygon formed by CHs (if a network comprises only one CH, than the central point is the CH itself).

5.1.2 Throughput Maximization

On the contrary, when a traffic is high, there will be a lot of collisions. All SNs are transmit-only, hence they cannot sense the existing traffic and avoid collisions. In this regime, CHs should concentrate only on high-priority SNs in their neighborhood and maximize the total number of packets they capture from these types of SNs. Thus, in high-traffic regime, our performance metric is **throughput maximization.**

Note that we do not explicitly consider issue of fairness in case of the high traffic scenario. Potentially, it can happen that we maximize total capture rate by receiving packets only from very close SNs and ignore the distant SNs. However, by our design requirements, high priority SNs are always close to the CHs, and they will all have similar priorities even using this metric. Only low priority SNs will starve, which is a trade-off we have already accepted in system requirements.

5.1.3 Fairness Metrics

As a future work, we plan to evaluate more metrics that will reflect the fairness issue. The first one is **proportional fairness** [3]. Each SN is assigned a log utility which is log of the number of successfully transmitted packets. The goal is to maximize the sum of log utilities of all SNs. This metric is widely used in networking. The second one is α**-coverage time**. It is the time until at least one packet is received from α fraction of deployed SNs. It depicts the capability of the CHs to extract information from the whole network.

5.2 Range

In this section we give numerical results on range maximization. We consider a setting with one SN and 1, 2 and 4 CHs, as depicted on Figure 5. We look at the fraction of packets captured by the CHs for various distances d.

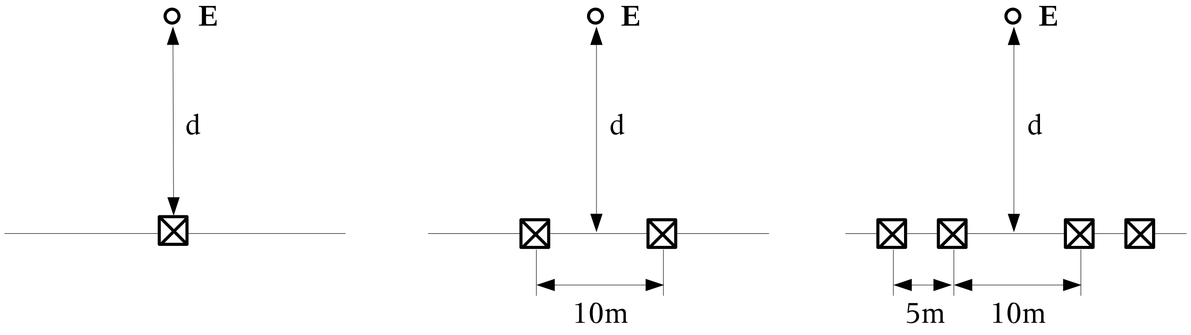

Figure 5: Topologies for measuring range: 1, 2 or 4 CHs are placed on a line. A SN is placed at a distance d from the line. We look at the fraction of packets captured by CHs for various d.

As explained in Section 2, we use pulses of 30 pJ. In Figure 6 we show the fraction of captured packets for uncoded transmissions, assuming the CS uses maximum ratio combining. We see that a single CH can achieve ranges of up to 30m. However, 4 CHs can increase the range up to 55m.

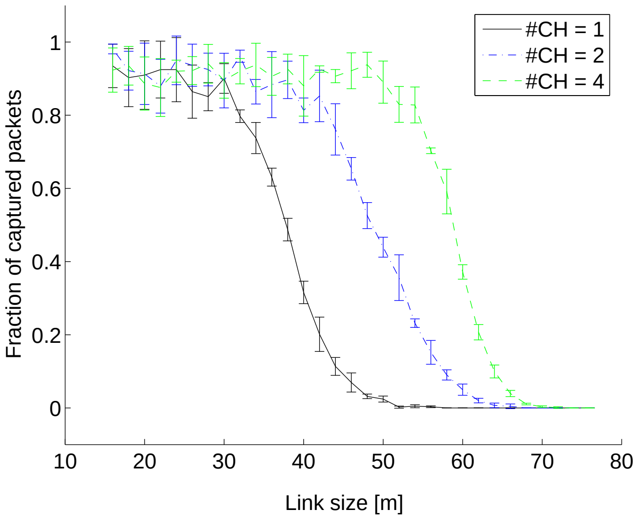

Figure 6: Fraction of captured packets for different SN's distances (scenario from Figure 5). We fix pulse energy to 30 pJ and use no coding. At the CS side we use maximum ratio combining.

In Figure 7 we fix SN distance $d = 60$m and we vary the pulse energy. We see that considerable energy can be saved by increasing the number of CHs and using combining at the CS.

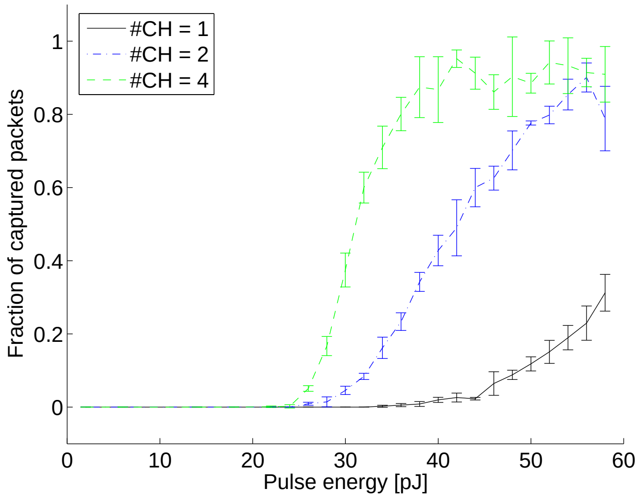

Figure 7: Fraction of captured packets for different pulse energies (scenario from Figure 5, $d = 60$m). We use no coding and at the CS side we use maximum ratio combining.

Finally, in Figure 8 we fix the distance $d = 60$m, pulse energy to 30 pJ and we vary code rate (code rate p means the code is capable of sustaining error probability p; see Section 2.3 for more details).

We see that when we have one CH, coding can significantly increase the range. However, when we have 4 CHs, the improvement with coding is very small.

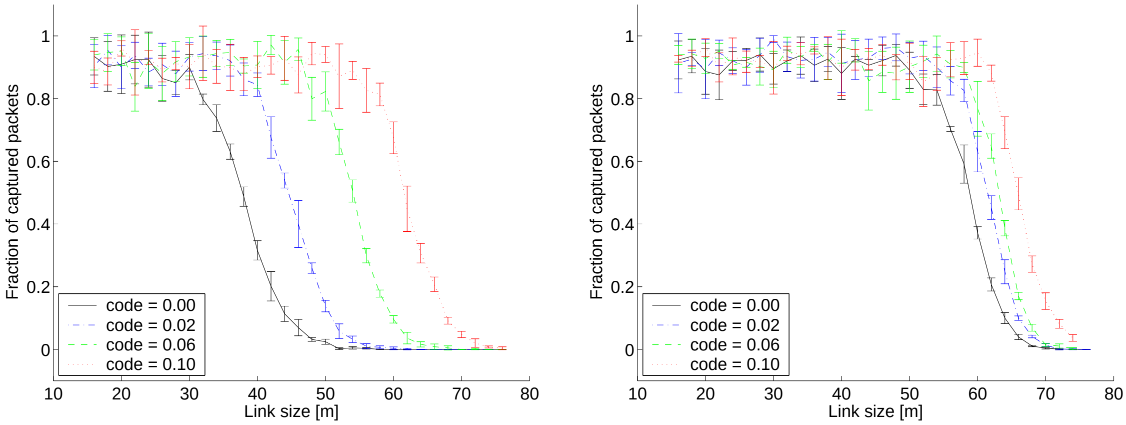

Figure 8: Fraction of captured packets for different coding and numbers of CHs (scenario from Figure 5, $d = 60$m, pulse energy 30 pJ). We use use maximum ratio combining at the CS side. On the left is the case with 1 CH and on the right with 4 CHs.

We conclude that to achieve the range of 60m we have to use 30 pJ pulses and either coding or combining with multiple CHs. We also see that a use of both coding and combining does not additionally improve the range. As we will see later in this section, a use of coding decreases total throughput. Therefore, we opt for using no coding and to improve range by putting a sufficient number of CHs.

5.3 Total Throughput

5.3.1 Random Networks

We first analyze random networks with 20 and 50 SNs uniformly distributed on a 40m × 40m square (note that all SNs are within range). We let them all have the same packet rate. We put 1, 2 or 4 CHs, as depicted on Figure 9. We vary the packet rate and observe the rate of captured packets.

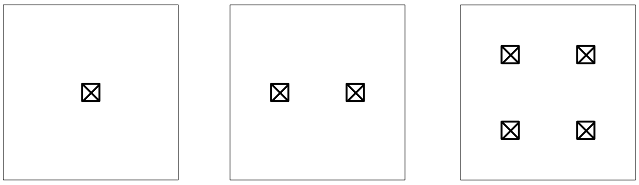

Figure 9: Static case: SNs are uniformly distributed on a 40m × 40m square, with 1, 2 or 4 CHs positioned as depicted on the figure.

CH architectures: The results are depicted on Figure 10 - 13. When the total load is small, there is no improvement due advanced CH architectures, since it is optimal to detect and receive packets from all SNs. We also see that in this case coding even degrades the performance. The use of coding increases packet sizes (in this case roughly by two), while allowing only 10% of error rate.

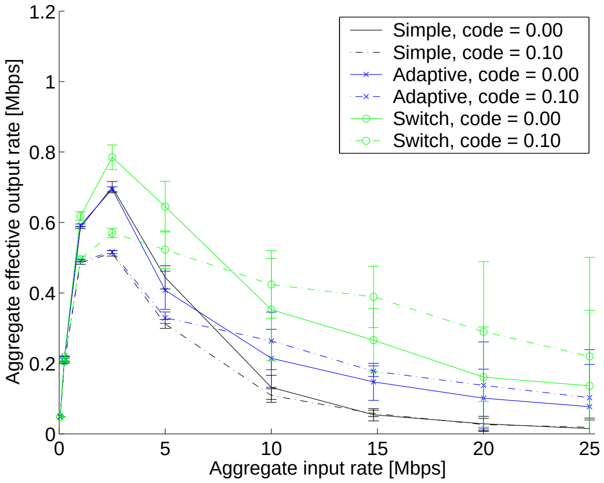

Figure 10: A scenario with 50 SNs and 1 CH: All SNs are assumed to generate Poisson traffic with the same intensity. On the x-axis we plot the aggregated input SN traffic, and on the y-axis we plot the aggregate goodput.

When the total load increases above 10 Mbps, the differences in performance between the different CH architectures become significant. The adaptive threshold architecture performs

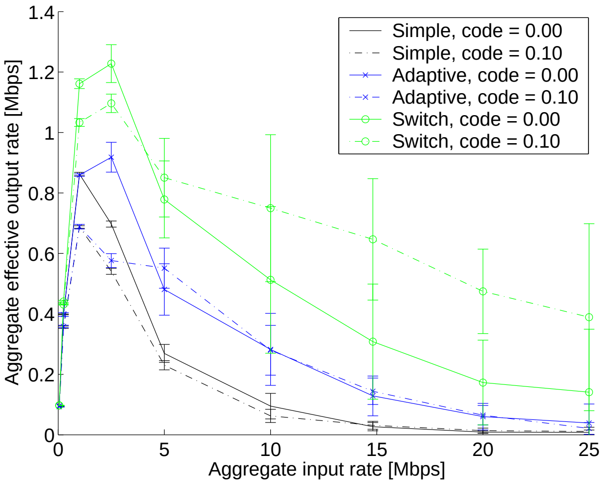

Figure 11: A scenario with 50 SNs and 2 CHs with no combining: All SNs are assumed to generate Poisson traffic with the same intensity. On the x-axis we plot the aggregated input SN traffic, and on the y-axis we plot the aggregate goodput.

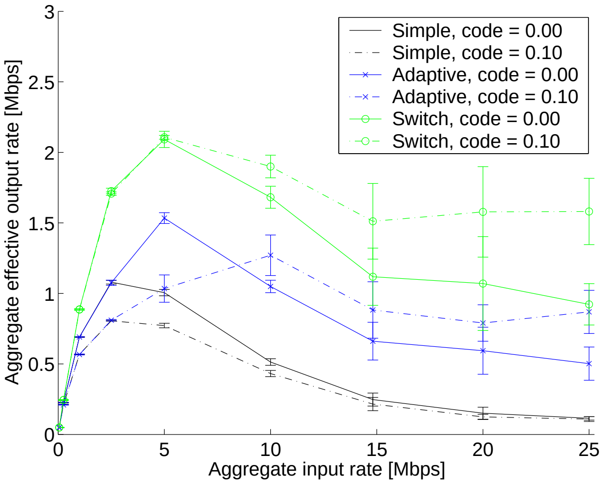

Figure 12: A scenario with 50 SNs and 4 CHs with no combining: All SNs are assumed to generate Poisson traffic with the same intensity. On the x-axis we plot the aggregated input SN traffic, and on the y-axis we plot the aggregate goodput.

twice as good as the simple architecture with no threshold. Also, the switching architecture performs roughly twice as good as the adaptive architecture.

Another interesting conclusion can be made about coding. For the simple CH architecture coding does neither improve nor degrade performance. For the adaptive threshold architecture, coding slightly improves the performance, while for the switching architecture coding drastically outperforms the no-coding approach.

So, the phenomenon is that coding improves performance for very high packet rates; on the contrary, when rates are low, coding makes things worse.

Let us first consider the low traffic case. Then, collisions will mainly occur because two strong packets overlap, since the probability of having more than two packets overlapping is

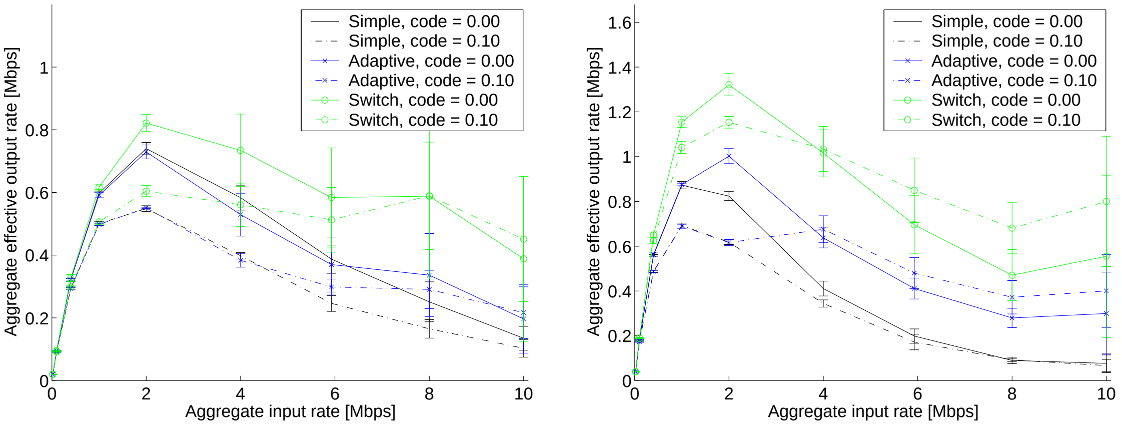

Figure 13: A scenario with 20 SNs. All SNs are assumed to generate Poisson traffic with the same intensity. On the x-axis we plot the aggregated input SN traffic, and on the y-axis we plot the aggregate goodput. On the left is the scenario with 1 and on the right with 2 CHs.

small. The overlapping is thus highly correlated: if two packets overlap at time t, then they will likely overlap at time t + delta (since both packets are being transmitted). In other words, if packets overlap, they will likely overlap for a period of time longer than 10% (which is the error resilience of the maximum code), and coding will not help at all. Instead, it is better to use no coding, have shorter packets, and try to avoid collisions altogether. In case of switching architecture, a collision with a stronger packet is completely avoided by CH design, hence there is no decrease of performance if we code, as can be seen in Figure 12.

Next, let us consider high traffic scenario. In the case of adaptive architecture, we have small detection region. Packet from detection region will still have relatively low rate (75%), and there will be many more packets from outside of the exclusion region. Now these weak packets are more likely to cause collisions. But since these packets are distant, it is not sufficient to have one interfering packet; we need to have several weaker packets overlapping with a strong packet to cause collision. Now this interference is much less auto-correlated as it is composed of a large number of interfering packets. In this case, coding can successfully cope with errors.

If the case of switching architecture, all collisions with a stronger packet will be eliminated, and all collisions will be caused by a large number of weak packets. Therefore, the improvement with coding is even higher.

Note that for example in Figure 13 there is an increase of the number of captured packets with the increase of traffic when the traffic is high. Similar phenomenon can be observed in Figure 12. This is an artifact of random topology generation during the simulations. Namely, we first select traffic rate, then create 5 random topologies, and for the same topologies vary the other parameters. However, for different values of traffic load we created 5 different random topologies. Therefore, this increase is artificial and comes from the fact that the topologies used for 10 Mbps loads on the figure on the right had an inherently higher throughput due to better SN positioning. It is thus a good idea to rerun simulations such that the same 5 topologies are used for all different traffic loads, in order to avoid these artificial variations.

Combining: We next consider the effects of combining. We have seen that in the low-traffic case combining at the CS can increase the range of a network. In order to analyze the effects of combining in high-traffic cases, we consider random networks with 50 uniformly distributed SNs in a (90m × 90m) square, and 2 CHs, placed as shown in Figure 9. We compare the effect of combining in conjunction with different codes and CH architectures. We use a larger area than in the previous cases. This way in most cases we have some SNs in the corners of the square that are outside of reach of a single CH and can be received only by combining. The results are depicted in Figure 14.

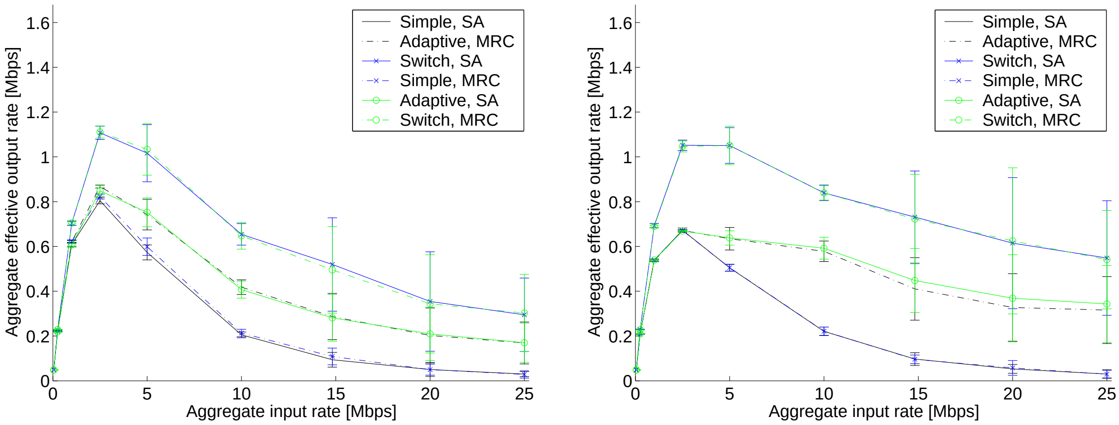

Figure 14: Fraction of captured packets with and without combining. Scenario from Figure 9: 90m × 90m square, 50 SNs, and 2 CHs. On the left is the case with no coding and on the right with 10% error code.

We see from the results that combining does not help at all in the high-traffic case. This is expected, since packets from distant SNs are anyway lost in case of high-traffic, and combining does not improve reception from nearby SNs. Moreover, combining does not decrease the performance, regardless of chosen SN or CH architectures, hence it can be safely used in any network design to increase the range. The results presented here are for networks with 50 SNs and 2 CHs, but we obtain similar results for networks with 4 CHs and/or 20 SNs.

5.3.2 Detecting Weak Packets in Presence of Strong Packets

As we have discussed in Section 1.5 and Section 5.1, for high data rate applications, it is required that high-priority SNs are placed close to a CH. In this section we analyze what happens when this requirement is not fulfilled. To that end, we construct a special example of a network, showed in Figure 15.

All SNs transmit data with the same rate. However, SNs in the lower left corner are detecting a fake event. The CS discovers this and drops all the packets coming from that region. Therefore, we count only packets coming from SNs from upper right corner. Packets from lower left corner are thus pure interference, whose signal is stronger then the signal of data packets. What is important is that we do not particularly adapt the architectures of SNs or CHs to this situation. We take the proposed architectures and we analyze how well they can handle this degeneric case.

The results are depicted in Figure 16. We see that the goodput is twice as small than the one shown in Figure 13 on the left (which represents a similar scenario with 20 SNs, 2 CHs and a larger square). We also see that the maximum is reached when the aggregate load is approximately 4 Mbps (unlike in the other example where it is reached for 2 Mbps). This is

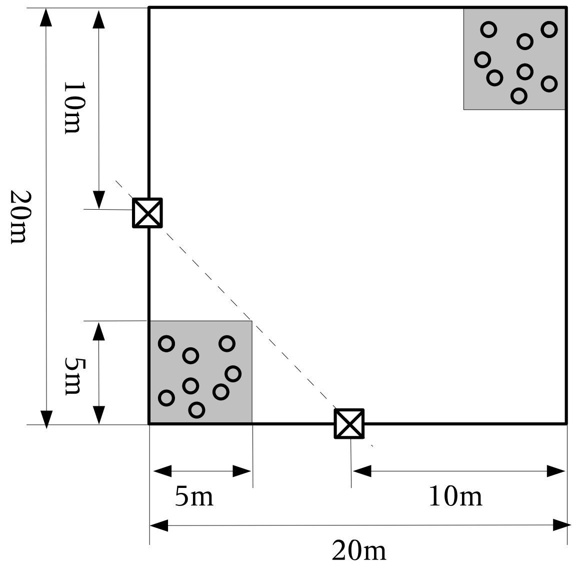

Figure 15: A special example of a network that tests the performance of a network when high-priority SNs are not well covered by the CHs. We consider a (20m×20m) square. There are 2 groups of 10 SNs, one in the lower left and one in the upper right corner (shaded areas on the figure). SNs in the lower left corner detect a fake event and CHs are not interested in their packets. We count only packets detected from SNs from the upper right corner.

the case since the example from Figure 15 can be roughly viewed as a network with 10 nodes in the upper right corner where some packets are randomly deleted (during the interference).

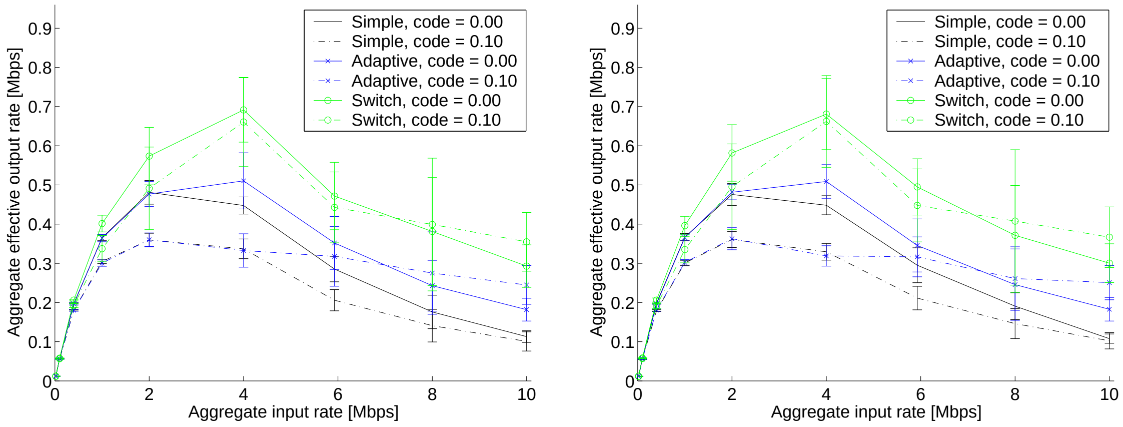

Figure 16: Numerical results for the scenario from Figure 15. On the left is the case with no combining and on the right with maximum ratio combining. On the x-axis we plot the aggregated input SN traffic (including sensors from both corners), and on the y-axis we plot the aggregate goodput (considering only SNs from the upper right corner).

Another interesting thing that can be observed from Figure 16 is that switching is still better than adaptive threshold, which is still better then no threshold. However, the difference is smaller. The explanation is that in most of the cases the advanced CH techniques successfully detect when a stronger packet from a lower left node would destroy a data packet from an upper right node, so the pre-selection of packets does not penalize the performance. On the contrary, advanced techniques still help in preselecting data packets coming from upper right corner, thus they still increase the performance in case of high traffic. This superficial analysis only offers a first effort to understanding the problem. It remains as a future work to do a more thorough analysis of the above emphasized phenomena.

Finally, we can also verify, similar as it is done in Figure 14, that combining in this case does not bring any improvement. However, the size of the square is rather small. We suspect that combining would improve the performance if SNs that transmit data packets are sufficiently far away, and it remains to be verified as a potential future work.

5.4 Dynamic Behavior of the Adaptive Algorithms

The goal of this section is to show how fast the threshold adaptation algorithm can adapt to a change in network traffic. We analyze a dynamic scenario, depicted on Figure 17. The framework for the scenario is defined in Section 1.5. We assume that the traffic generated during measurements is 830 kbps and otherwise is 1kbps (for sake of visualization; otherwise it could also be 0).

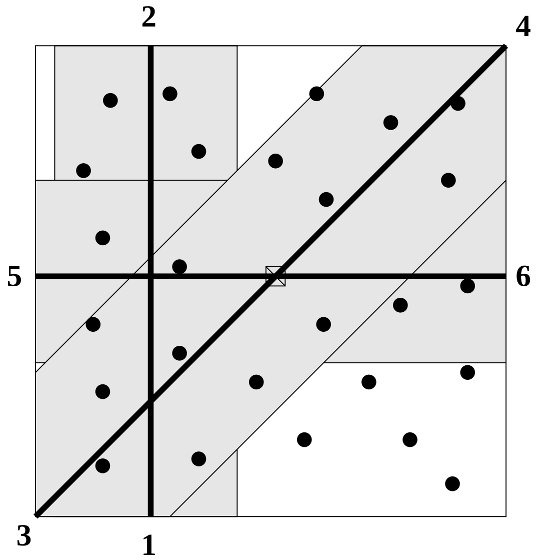

Figure 17: Dynamic case: 50 SNs are uniformly distributed in a (40m×40m) square, with one CH in the center. An event is generated on line 1-2 at 0.5s and lasts 0.1s. It is detected by SNs that are 8m far away from the line. An another event occurs on line 3-4 form 0.6s to 0.7s and the third one on line 5-6 from 0.65s to 0.8s. When a SN detects an event it starts sending data with rate 830 kbps. Otherwise it sends only 1kbps.

Thick lines represent events, and shaded boxes represents the areas in which SNs can sense these events. Event 1-2 occurs from 0.5s to 0.6s, event 3-4 from 0.6s to 0.7s and event 5-6 from 0.65s to 0.8s. We assume that the entire lines are activated during the denoted periods. The goal of the CH is to collect as many information as possible from the SNs at all times. When the traffic is high, it will decrease the detection region and will capture more information from a smaller surface. Otherwise, it will try to detect packets from a wider area.

The results of simulations are depicted in Figure 18 and Figure 19. On the left of Figure 18 we see the rate of captured packets and on the right we see the value of the adaptive detection

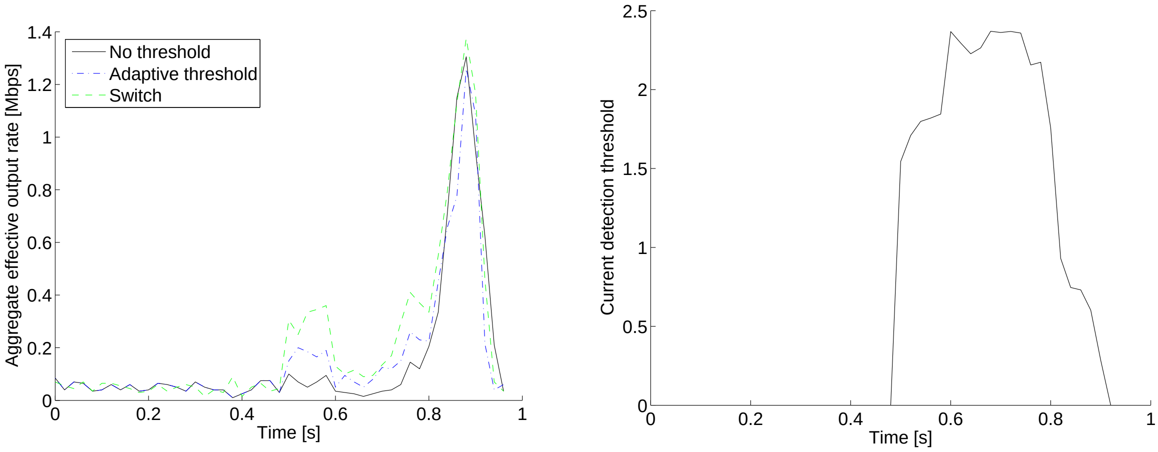

Figure 18: On the left, we give rate comparison of the two approaches applied on the scenario from Figure 17. On x-axis is time, and on y-axis are achieved throughput. On the right we depict how the threshold is adapted as the traffic varies, for the same scenario. On x-axis is time, and on y-axis is the threshold in SNR (minimum SNR of the packet needed to be detected).

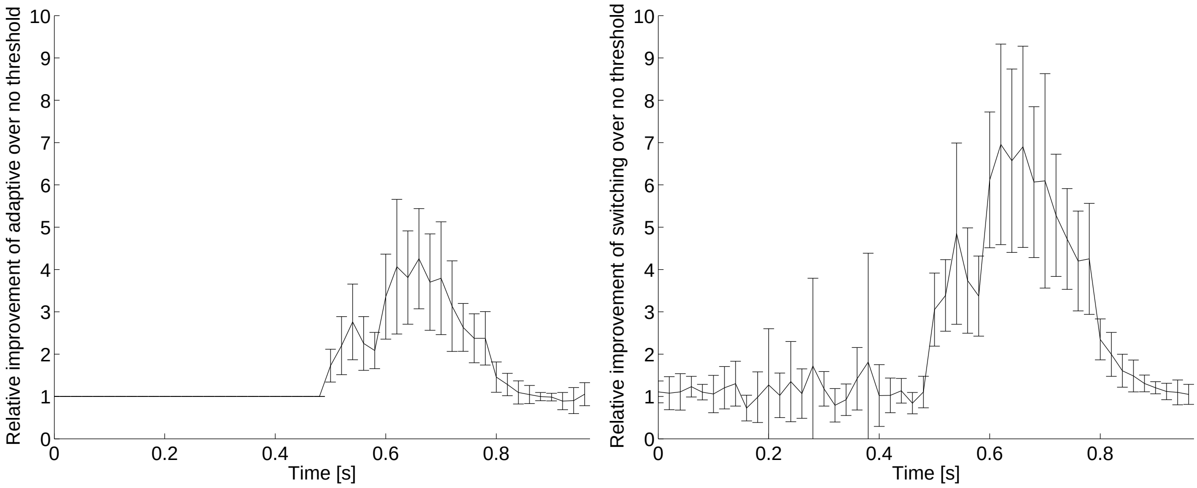

Figure 19: On the left we see relative improvement of adaptive threshold over no threshold approach and on the right we see the relative improvement of switching over no threshold approach. On x-axis is time and on y-axis is the relative improvement.

threshold. The rate increases around 0.5s when event 1-2 occurs. At that time the switched architecture is twice better than the one with adaptive threshold, while the latter is twice better than the one with no threshold, as can be seen in Figure 18. During that time, the adaptive threshold increases.

At 0.6s, two additional events occur, and the rate drops because the system is saturated with too much traffic. While the rates drops regardless of the CH architecture, the ratios stay similar. The same pattern is observed until the end of the simulation.

6 Conclusion

In this report we analyzed different SN's, CH's, and CS's architectures for transmit-only wireless sensor networks.

We first considered SN design. We derived the power of a pulse such that the range of communication satisfies the application requirements. We showed that in most of the cases the use of coding deteriorates the performance. And we proposed a power-aware, pseudo-random scheduled access scheme for the SNs, which allows the CHs to turn off their receivers during the idle periods.

We next analyzed three different CH architectures: no-threshold architecture, adaptive-threshold and switched architecture. Adaptive-threshold architecture requires the same hardware as no-threshold architecture, therefore has the same cost. Switched architecture introduces an additional detection and synchronization circuit. It is thus expected to be more expensive.

We find by numerical simulations that the three CH architecture perform equally good for low traffic. For high traffic, the switched architecture is twice better than the architecture with adaptive threshold, and the architecture with the adaptive threshold is twice better than the one with no threshold.

Therefore, adaptive-threshold architecture brings a significant performance increase during high-rate traffic bursts and can be implemented with no additional costs. If this is not sufficient, an additional improvement can be introduced through the switched architecture, at a slightly increased cost of a CH.

Finally, we analyzed different CS architectures. We found that in case of low traffic, combining techniques can significantly improve the range of communications. We also found that in the high-traffic case combining does not improve but also does not deteriorate the performance.

The design presented in this paper is performed for networks with transmit-only SNs. The presented numerical results are based on physical layer described in [7]. Nevertheless, the ideas are applicable to other physical layers as well, where it is prohibitively expensive to put a receiver in each SN.

References

[1] J. Barry, D. Messerschmitt, and E. Lee. *Digital Communication: Third Edition*. Kluwer Academic Publishers, 2003.

[2] T. Cover and J.A. Thomas. *Elements of Information Theory*. John Whiley & Sons, 1991.

[3] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[4] B. Radunovic. A cross-layer system simulator for UWB-based wireless sensor network. *IBM Research Report RZ 3594*, February 2005.

[5] D. Rus. Keynote on autonomous mobile networks. In *The First IEEE Workshop on Embedded Networked Sensors (EmNetS-I)*, 2004.

[6] B. van der Wal et al. Definition of UWB scenarios, Deliverable D2a2 - Initial. *Integrated Project PULSERS, http://www.pulsers.net*, March 2004.

[7] M. Weisenhorn. Physical layer for reader scenario. *IBM Research Report RZ 3595*, 2005.

[8] M. Weisenhorn and W. Hirt. Novel Rate-Division Multiple-Access for UWB-Radio-Based Sensor networks. In *Int. Zurich Seminar on Communications (ISZ)*, 2004.