Data filtering and aggregation in a localisation WSN testbed

I.F.R. Noppen[2], D.C. Dimitrova[1], and T. Braun[1]

[1] Universität Bern, Bern, Switzerland,
{dimitrova|braun}@iam.unibe.ch
[2] Universiteit Twente, Enschede, The Netherlands,
i.f.r.noppen@alumnus.utwente.nl

Abstract. The main challenge in wireless networks is to optimally use the confined radio resources to support data transfer. This holds for large-scale deployments as well as for small-scale test environments such as test-beds. We investigate two approaches to reduce the radio traffic in a test-bed, namely, filtering of unnecessary data and aggregation of redundant data. Both strategies exploit the fact that, depending on the tested application's objective, not all data may be of interest. The proposed design solutions indicate that traffic reduction as high as 97% can be achieved in the specific case of test-bed for indoor localisation.

Key words: WSN, filtering, aggregation, WiFi, bluetooth

1 Introduction

Wireless sensor networks provide excellent means for monitoring and data gathering in a large range of application areas. One such application is the use of radio-enabled sensor nodes for (indoor) positioning in which the sensor nodes collect signal measurements of user devices using radio transmissions, e.g., Bluetooth. Among the most frequently used radio standards are the IEEE 802.11 (with commercial name WiFi) and Bluetooth standards. Processing of the collected measurements can derive the location coordinates of the transmitting device. Potential use cases of a positioning application include, but are not limited to, analysis of visitor behaviour in shopping malls, tailored discount dissemination in attraction parks and evaluating staff efficiency in hospitals.

Inspired by the many use case opportunities, the Location Based Analyser (LBA) project addresses the indoor localisation challenge by leveraging radio frequency (RF) based technologies, namely WiFi and Bluetooth. More specifically, we use multiple sensor nodes at known positions to collect measurements on the received signal strength indicator (RSSI) from personal devices on the premises. The collected measurements are periodically sent to a central database server where they are sorted per observed device and processed to determine the current position of each device. There are various techniques to map RSSI to distance, the most often cited being (multi)lateration and fingerprinting.

As part of the development process of the localisation system we set up a test-bed for the purpose of testing and performance evaluation. Early in the design and testing phase we stumbled across the problem of rapidly growing sensor data. Although test-beds are designed typically at a smaller scale than the finally deployed system, challenges related to congestion of the wireless medium may arise. In addition, data storage may prove another affected aspect. In order to ensure non-disrupting operation and system scalability one needs to take care when managing the radio resources. We chose for an intuitive approach that classifies the wireless traffic and identifies what data is pertinent for the needs of the application. Possible data reduction strategies include filtering of unnecessary data, aggregation of redundant data and data compression. Data aggregation in WSNs has been largely studied [5,9,12] and evaluated in WSN testbeds [1,3,10] for the purposes of reducing traffic volume and energy consumption.

This paper describes how we adopt filtering and aggregation to minimise wireless traffic in a test-bed. Contrary to other studies, e.g., [9], which address hierarchical aggregation in the network, we are only interested in a local (on a single node) aggregation. We take as an example the case of a localisation application but the discussed data reduction strategies can be applied to a larger set of applications by modifying parameters of switching functionality on or off.

The rest of the paper is organised into the following sections. Section 2 describes the specific test-bed that we use and the encountered traffic challenges, given the application's needs. Next, Sections 3 and 4 discuss the implementation and performance of filtering and aggregation respectively. Their combined use is analysed in Section 5. Finally, Section 6 summarises the paper.

2 Localisation WSN test-bed

2.1 Localisation of user devices

We designed an indoor localisation system that relies on sensor nodes at known positions, which collect signal measurements from personal devices. A central server processes the collected measurements to derive the location of the personal devices. We are interested in signals from personal devices and in signals from the sensor nodes. These signals are used to monitor the quality of the radio channel and to improve the performance of the system.

In order to test and evaluate the localisation system we built a test-bed inside a single room, which reflects the system design. Its objective is to collect measurements of signals from personal devices and reference sensor nodes. The test-bed contains 16 sensor nodes, which form a 4x4 grid at 0.5 meter below the room ceiling, see Figure 1. The sensor nodes scan continually for WiFi and Bluetooth signals and record the RSSI levels. Periodically, each sensor node sends its measurements to a gateway node, which collects all measurements and forwards them to the database server. At the server the measurements are stored and analytically processed. In the rest of the paper the term sensor node and sensor are used interchangeably.

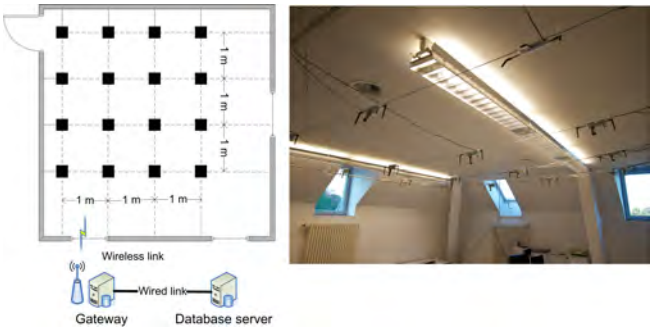


Fig. 1. LBA test-bed architecture: multiple sensor nodes (SNs), connected over WiFi to a gateway (GW). Measurements are stored in a database server (DB)

In our proposed localisation system the amount of wireless traffic depends on the number of deployed sensor nodes, the frequency at which measurements are reported, and on the number of detected or tracked devices. Additionally, there is interfering traffic from other devices that may use the same wireless medium. Since the test-bed is located in the Computer Science building of the University of Bern, there are other experimental wireless networks and wireless access infrastructures that use the same radio channels. We consider traffic from such networks and WiFi access points, e.g., beacons, as non-informative since measurements regarding these devices contribute neither to the localisation of devices nor to the radio channel monitoring. Therefore, these are unnecessary measurements and while we cannot always avoid their collection we can prevent their transmission to the central server. To this end we apply filtering.

Another factor that directly affects the busyness of the radio channel is the proportion of collected measurements per device. For each device several measurements may be collected by one sensor while a single measurement would suffice for localisation. Clearly, reporting all measurements would be redundant and we would like to minimise the radio channel utilisation. This can be achieved by decreasing the amount of data to send. Therefore, we explore the use of data aggregation.

2.2 Test-bed implementation

As sensor nodes we used Gumstix Overo Fire devices [2] with integrated on-board Bluetooth and WiFi interfaces, which support the scanning of the wireless medium. Moreover, we attached an additional WiFi card, which is used for communication between the sensor node and the gateway. On the sensor nodes, we run a light Linux kernel and several lightweight packages to keep the sensor as lightweight as possible. This sensor software is built using the Administration and Deployment of Adhoc Mesh (ADAM) framework [11] which includes the custom packages needed for running the Bluetooth and WiFi scanners. WiFi measurements are collected by capturing packets on the wireless interface with

libpcap, an application programming interface for capturing network traffic, and by hopping through the WiFi channels with an interval of one second. Bluetooth measurements are recorded by using output from the *bluez* Linux Bluetooth library. When the measurement buffer is full or a predetermined period has ended, the measurements are flushed to the gateway. All the sensor nodes and the gateway are situated in the same local network.

The gateway and the database server both run on a regular Linux distribution. The gateway can reside on both a desktop machine or a sensor node, as long as there is a wired connection available for the communication to the database server. The communication is through a SOAP web-service. At the database server, we use a regular MySQL databases.

3 Filtering

One strategy to decrease traffic in the test-bed (and any other wireless network) is to transmit only data that is pertinent. Filtering is a method that can successfully omit the collection and transmission of unnecessary measurements. The choice of an appropriate filtering solution needs an answer to three *design questions*: what to filter, where to filter and how to filter.

What to filter A filtering solution is needed that can identify unnecessary measurements and only allow the transmission of measurements on user devices (used for localisation) and reference sensor nodes (used for channel estimation). As discussed earlier, we consider measurements on signals from experimental wireless networks and WiFi infrastructure, e.g., access points (APs), as unnecessary. We refer to the former group as 'always-on devices' and to the latter - as 'fixed infrastructure'. Each group requires different filtering strategies as it is explained later.

Where to filter Three places can be identified in our system where we can employ filtering: the sensors, the gateway and the database server. Filtering at the sensors has a direct impact on buffer occupation and on wireless traffic but is challenging due to their limited resources while the decision what to exclude needs large sets of measurements. The only benefit of filtering at the gateway is the decreased traffic towards the database server. However, often bandwidth is not a problem since wired connections are used generally. Moreover, the gateway also does not have knowledge on long-term data. On the database server, we have both the capacity and the measurements at our disposal to support a decision making for filtering. Hence, it is a more appropriate system for the filtering decision process.

How to filter Filtering can be based on black- or whitelisting of certain MAC addresses. When a certain MAC address is blacklisted, all measurements of that MAC address are discarded. When a MAC is whitelisted, all measurements related to it are collected. The choice of strategy depends entirely on the application's objective. In a controlled environment, when the target group of devices to monitor is well defined, whitelisting is the better choice since we are only interested in measurements from a limited set of known MAC addresses. In realistic

environments, where we have no control over the target devices, whitelisting is not feasible because the targeted MAC addresses are not known beforehand. In such case blacklisting is the better choice.

Another classification criterion is how the decision what to filter is taken. If we collect the MAC addresses and enter them manually into the filtering system we call this *static filtering*. Static filtering is time consuming, requires effort and does not scale well. A better alternative is *dynamic filtering*, which introduces certain intelligence in the system. Such a system integrates decision making processes to analyse incoming measurements and decides what MAC addresses to filter out.

3.1 Filtering solution

Taking into account the requirements of the current experimental test-bed we chose a dynamic blacklisting strategy with static elements and static whitelisting support, which we term combi-listing.

Static blacklisting Static blacklisting refers to the filters that are directly installed at the sensor to filter out signal measurements from the fixed infrastructure (APs). An AP contributes significantly to the wireless traffic because (i) it typically sends a beacon message every 100ms and (ii) it serves multiple clients in parallel. Note that measuring this kind of traffic is undesirable, independently of the specific WSN application.

Filtering of the fixed infrastructure is quite easily done at the sensor nodes using the two distribution system (DS) flags in a WiFi packet [4] that indicate sender (first bit) and receiver (second bit): 0 for mobile device and 1 for AP. Hence, since they already use *libpcap* to capture packets at the WiFi interface, we only need to create an additional rule to discard all packets with the type 'Beacon' or DS flags 10 or 11 (first bit 1 indicates AP originating traffic). Static blacklisting is implemented using the existing *libpcap* functionality.

Dynamic blacklisting Static blacklisting on top of *libpcap* is not feasible for the identification of always-on devices that behave as any other device but are active continually or for long periods of time. Instead we use a dynamic blacklisting technique that combines a decision making process, which periodically generates blacklists, and a dissemination process, which distributes the blacklists to the sensor nodes.

Decision making The decision making process is situated on the central server and is responsible for the generation of the blacklists - one for each sensor node. The process relies on one commonality between all always-on devices, namely, they are generally connected 24/7. Therefore, if we analyse the collected measurements over a long period we should be able to identify always-present MAC addresses that correspond to always-on devices.

Formally a device in our test-bed can be identified by its MAC address and *activity level*, i.e., the percentage of time in which measurements of its MAC address were received. The activity level is calculated over a specific *evaluation*

period, which is the timespan over which the list of blacklisted MAC addresses is generated. For example, if a device was active for two hours within an evaluation period of eight hours it has an activity level of 25%. If we define an activity level threshold and a device's activity level is above this level we can deduce that this is an always-on device. The choice of the threshold is very important and related to the duration of the evaluation period. For instance, it is fair to say that a threshold of 80 or 90% should allows the identification of always-on devices.

An easy way to implement the proposed decision making is to count the number of distinct timestamps for a specific MAC address and divide this by the total number of seconds in the evaluation period. However, there are disadvantages to that in our test-bed. First, our WiFi-scanner hops channels every second. Second, if an always-on device is only connected to the network and not actively transmitting it will have only few measurements. To correct for this, we divide the evaluation period into equal-length *activity periods*. Per MAC address, we check within each activity period whether there is at least one measurement of this address. If this is the case we mark the period as true, otherwise we mark it as false. If we now count the number of activity periods marked as true and divide that by the total number of activity periods in the evaluation period (equation 1) we will get the percentage of time that this MAC address has been active. We can derive the number of activity periods by dividing the evaluation period by the activity period, both measured in seconds.

A simple comparison of the activity level threshold with the activity levels of all MAC addresses detected within the evaluation period will give us the MAC addresses to include in the blacklist.

$$ActivityLevel = \frac{count(ActivityPeriod\_True)}{EvaluationPeriod/ActivityPeriod} \tag{1}$$

Dissemination The dissemination of blacklists is pull-based. The procedure is shown in Figure 2. The sensors request the blacklists from the central database server via the gateway node. The server can answer, also via the gateway, either with a new blacklist, when available, or with an empty message, when the sensor polled too early and no update is available yet. Note that the new blacklist from the server can contain no MAC addresses when there are none to filter out. If a sensor should blacklist certain MAC addresses it filters out their measurements but keeps statistics for each of the blacklisted address. Periodically this information is sent back to the server, where it is used to re-evaluate whether the MAC address should stay blacklisted. Without these statistics the decision support process will loop into a repetitive adding and removing of MAC addresses to the blacklist.

The dissemination procedure is implemented by extending the test-bed functionality and introducing three new message types, namely, blacklist request, blacklist update and blacklist aggregate messages. Either on start-up or after a timer expires, the sensor nodes request a blacklist through the gateway using a *blacklist request* message. The message contains a *timestamp* of the current

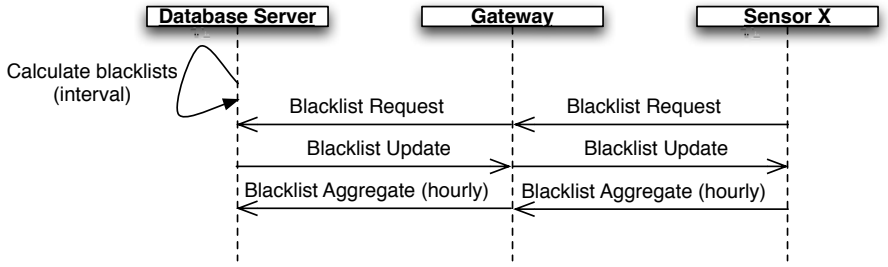


Fig. 2. Message exchange between sensors, gateway and database server regarding the blacklist

blacklist at the sensor and the *type* of blacklist the node is requesting (Bluetooth or WiFi).

The server answers with a *blacklist update* message, which contains the *timestamp* of the list in the update, or the timestamp of the blacklist request if no newer update is available. The *checkback-time* field suggests how many seconds will pass between the timestamp and the time a new blacklist will be available. The *list size* field tells us how many MAC addresses the complete blacklist contains. The *update* flag indicates if a new list is sent (flag 1) or if the sensor polled and there is no update (flag 0). When there are no MAC addresses to blacklist the flag is 1 (true) but the list size is zero and the MAC addresses field is empty. The length of the MAC addresses field for a non-empty list depends on the list size field.

Upon receiving the blacklist update message the sensor replaces the old blacklist with the new one and resets the timer according to the checkback-time field. For each blacklisted MAC address the sensor collects statistics and reports them hourly back to the server in a *blacklist aggregate* messages. The blacklist aggregates message contains one or more structures depending on the number of blacklisted MAC addresses. Each structure contains the MAC address of the blacklisted device along with the first-seen and last-seen timestamp, the number of measurements between the two timestamps and the average RSSI. The count is used to make a decision whether a MAC address has to remain blacklisted.

Static whitelisting Static whitelisting is used to ensure the collection of measurements on the reference sensor nodes which are used for channel evaluation in the test-bed. For that purpose the MAC addresses of all reference sensor nodes are identified and a specific whitelist for each sensor is kept at the central server. The server is responsible to check that a MAC address from the whitelist does not become blacklisted.

Alternative solutions The proposed distribution of node-specific blacklists uses many unicast connections, which may lead to depletion of radio resource if a large-scale sensor network is considered. Therefore, the realisation of the filtering solution may need modifications in order to scale down service traffic.

One possible approach is to broadcast the blacklist, that is the same for all sensor nodes in a specific area (the area size depends on the communication range of the radio technology). In addition, each sensor can pull its specific whitelist from the central server when coming online.

3.2 Experimental analysis

In this section we present results on the data reduction that filtering can bring but first we discuss some parametrisation issues.

Parametrisation Integrating the proposed filtering solution requires setting up some parameters such as the blacklist evaluation period and the activity period. For our purposes we selected an evaluation period of 24 hours, which aligns easily with human activity. Choosing a good value for an activity period is more challenging. In order to analyse this, we set up a test, where we included a fixed WiFi device (laptop) in idle mode in the test-bed. The device was only connected to a wireless network with no data traffic exchange. We let the sensor nodes collect measurements over 65 hours and calculated the activity level of the idle and the most frequently seen device at each of the sensor nodes for an activity period of 60 and 300 seconds. Corresponding box plots over all sensors are given in Figure 3.

A successful deployment should be able to filter out the idle device's MAC address as well as other high activity MAC addresses (most frequently seen device). As we can see in Figure 3, an activity period of 60 seconds will not lead to a successful identification of the idle device as 'always-on' since its activity level reaches only about 42% on average. When we change the activity period to 300 seconds the activity period of the idle devices rise up to 90% and it can be easily identified for blacklisting. The reason for the above behaviour is the idle status of the device in which case it communicates to the networks once every few minutes. Note that the most seen device is less vulnerable to short activity periods and easily reaches 80-90% of activity because it is actively transmitting.

Traffic reduction To quantify the gains in terms of reduced number of measurements we conducted the following experiment. First, the testbed ran for full 24 hours, after which both Bluetooth- and WiFi blacklists were generated for each node. Then, in a second 24 hours run no filtering was directly applied but the generated blacklists were used to calculate, for the same data set, what the measurement reduction would have been. This provides us a common base for comparison since we are using the same data set. In the filtering decision the parameters are: $activity\ level > 0.8$, $activity\ period = 300$ seconds, $evaluation\ period = 86400$ seconds (24 hours).

Table 1 provides detailed statistics on the measurement reduction per sensor node. The reduction is the percentage of measurements that will not be transmitted if using filtering. Interestingly, the size of the generated blacklist is rather small although the test-bed location would suggest much larger wireless activity.

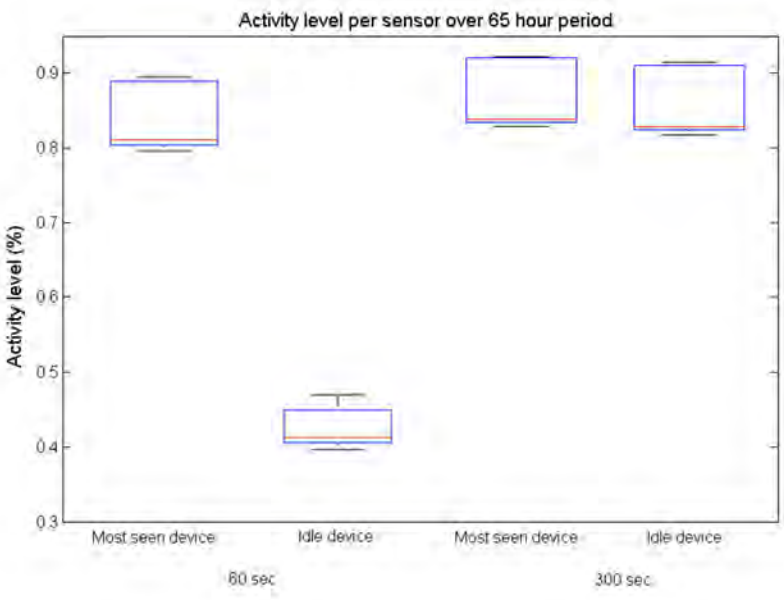


Fig. 3. WiFi: Activity level over all sensors for the most active MAC address and an idling device when the activity period is 60 seconds

We explain that with the fact that the experiments were conducted on a weekend when there are significantly less people, and hence always-on devices, in the building.

In terms of reduced measurement values the results show that the effect of filtering is significant. The reduction with combi-listing includes whitelisting of the MAC addresses of the other sensor nodes for reasons discussed earlier. For deployments where whitelisting is not needed the gains in reduction would be even bigger. This trend is better visible in Figure 4 for WiFi - the mean measurement reduction per sensor without whitelisting is about 93%, more than 10 percentage points higher than the mean reduction with combi-listing, i.e., the combined use of black- and whitelists.

For Bluetooth we registered even higher measurement reduction with 99.76% on average. We explain that with the smaller (six times) proportion of Bluetooth devices in our test-bed environment compared to the number of WiFi devices. As result one Bluetooth MAC address contributes more to the total number of measurements. In addition, the range of Bluetooth is smaller than for WiFi and therefore less devices will be detected in general. Note that whitelisting is not included because there are no addresses to be whitelisted for Bluetooth (we do not use Bluetooth signals in channel characterisation).

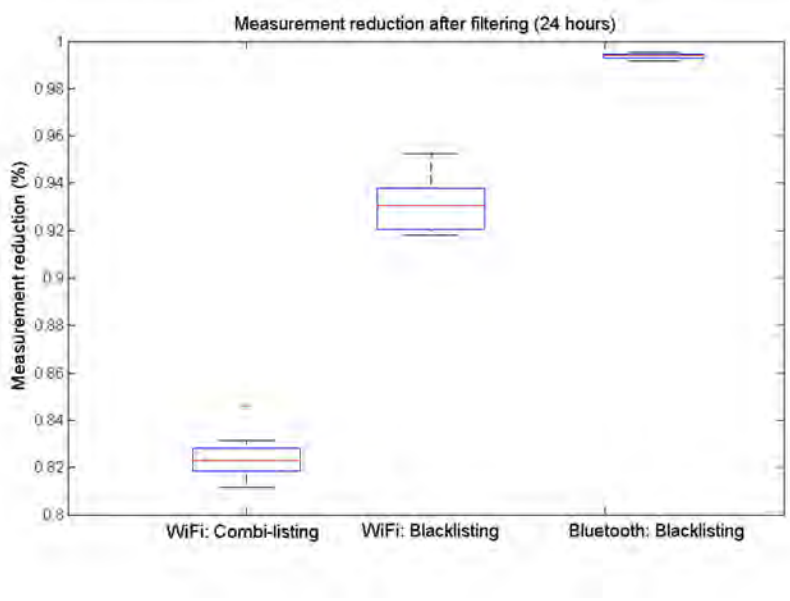


Fig. 4. Measurement reduction comparison over all sensors after filtering

Sensor	Total measurements	Reduced measurements	Reduction [%]	Blacklisted MACs
1	408.453	74.569	81.7%	26
2	458.547	83.204	81.9%	26
3	425.139	76.017	82.1%	26
4	416.985	72.839	82.5%	25
5	432.813	72.880	83.2%	27
6	404.987	71.233	82.4%	26
7	418.635	74.164	82.3%	25
8	412.666	76.420	81.5%	25
9	441.695	83.182	81.2%	26
10	390.836	70.235	82%	26
11	394.006	71.564	81.8%	26
12	441.728	78.060	82.3%	25
13	427.409	72.928	82.9%	26
14	364.902	62.817	82.8%	26
15	268.988	63.144	82.9%	25
16	414.818	63.914	84.6%	26

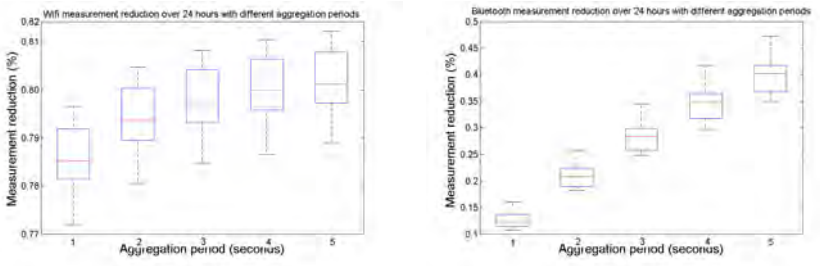
Table 1. Overview of WiFimeasurements in the second 24 hour period in the experiment.

4 Aggregation

Aggregation of data (measurements) is another strategy that can improve the utilisation of the limited radio resource and decrease the chances of collision. Generally speaking aggregation is a technique to decrease the amount of measurements sent over the wireless channel while retaining the measurements credibility. In sensor networks aggregation has been proposed to decrease energy consumption [8, 12] or network congestion [5]. We are interested in using aggregation to decrease network traffic and improve scalability since the current

deployment of electrically powered sensor nodes does not face energy consumption challenges.

4.1 Aggregation mechanism



(a) WiFi (b) Bluetooth

Fig. 5. Measurement reduction over different timespans of aggregation.

Several approaches towards data aggregation are possible. One strategy is to let the sensors report only changes in measured values, which is not suitable since RSSI vulnerable to external factors and not very stable. Another aggregation method is to send a single measurement (e.g., mean, max) per timespan where the timespan duration largely depends on the type of application. For example, for monitoring of ambient temperature one measurement per hour may be sufficient while for target tracking a timespan in the order of few seconds is more appropriate. We have chosen for the second option; the choice of timespan duration is investigated in Section4.2.

In addition, we need to select which value to report. In the case of RSSI we expect that the maximum value would be best since it is the least affected by propagation conditions. The feasibility of other choices such as an average value or another statistic registered over the aggregation timespan are discussed in another study, namely [6].

An alternative approach to decrease wireless traffic is data compression [7]. Instead of using the redundancy in measurements data compression gains from redundancy in the data itself by applying appropriate encoding. Although beneficial it also requires additional processing.

To enable the chosen aggregation strategy in the test-bed two buffers are set at the sensor nodes - one that collects all raw measurements and another for the reported values. When the first buffer is full, or at the end of a reporting period, all measurements are processed and the maximum RSSI per MAC address is written into the second buffer. Then, it flushes the data to the gateway.

4.2 Experimental analysis

To determine the measurement reduction we can achieve by applying aggregation, we used the same experiment setup as for the filtering experiment. We took measurements over the first 24 hours and calculated the measurement reduction if each sensor would apply aggregation. Given the mobility of the tracked device, we chose o aggregate over timespans of one to five seconds. We have chosen to report the maximum RSSI value since we believe they are least affected by propagation factors. The calculations were done for both WiFi- and Bluetooth signals.

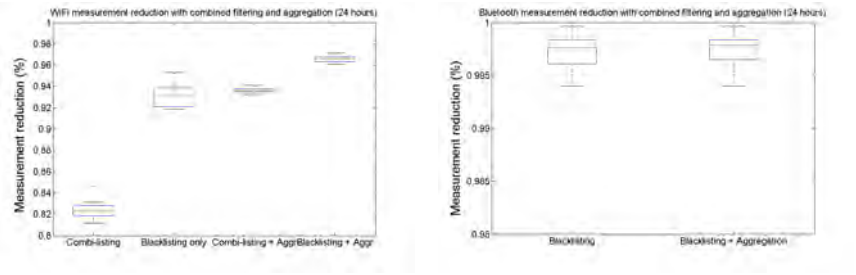
Results for WiFi are shown in Figure 5(a), where boxplots of the reduction achieved by each sensor node (y-axis) are plot against the used timespan (x-axis). As expected, aggregation significantly affects the measurement traffic and in our cases leads to a reduction in the number of measurements by more than 79% on average for a one-second timespan. Increasing the timespan to five seconds does only marginally improve the reduction to 81%. The reason for this is twofold. On the one hand, not all devices are broadcasting every second. On the other hand, the WiFi-scanner hops the wireless channels every second. Since devices communicate with a network on a single channel, we will not see their MAC addresses after this second again until we completed the cycle of channel-hopping.

In Figure 5(b) the results for Bluetooth show different patterns - the measurement reduction has an almost linear increase when we increase the timespan. More notably the measurement reduction grows from 12% to 40%, a less dramatic improvement than in the case of WiFi. The number of measurements we collect for Bluetooth are far fewer than the measurements collected for WiFi, reflecting the ratio of devices that use the two technologies. Based on the results we can conclude that the optimal value of the aggregation period for Bluetooth depends on the application needs.

5 Combined filtering and aggregation

While the individual measurement reduction of both filtering and aggregation shows great promise, it will be interesting to know if we can gain even more by applying both techniques in the same sensor network. To analyse this, we again used the results of the estimated measurement reduction experiment for the filtering and calculated the total measurement reduction when we apply aggregation (with one-second timespan) on top of that.

Figure 6(a) shows the results for WiFi measurements. For comparison reasons we include the results for filtering as shown in Figure 4. Aggregation adds an additional gain on top of the reductions that can be achieved by black- and whitelisting. Differences between blacklisting and whitelisting are consistent with previous observations - disabling whitelisting leads to even higher reduction. In our specific case, given we chose to apply a combi-listing, the combined reduction will be on average just shy of 94%.



(a) WiFi (b) Bluetooth

Fig. 6. Measurement reduction for combinations of blacklisting, whitelisting and aggregation.

The results of combining filtering and aggregation for Bluetooth are shown in Figure 6(b). For reference, the results on using only blacklisting are included. The graphs show that aggregation only slightly improve performance, the reasons being the few Bluetooth devices that the system detects and the efficient filtering of always-on devices that already brings measurement reduction of almost 99.8%. Although we are aware that the results are sensitive to the specific system deployment, we expect that aggregation will lead to smaller reduction in measurements for Bluetooth than for WiFi due to the typically lower number of Bluetooth devices. Note that there is no whitelisting for Bluetooth since in the current deployment it is not used for channel estimation.

6 Conclusion

This paper deals with radio traffic challenges arising in wireless sensor test-beds. We showed how the traffic volume can be greatly reduced by leveraging filtering and aggregation independently and combined. We achieved reductions of 80% on average with a peak above 95%, depending on used settings. Without this reduction in traffic, the testbed would not be able to scale well when extended to large testing sites due to the limited resources on the sensors and congestion of the radio medium.

The reductions were achieved for an experimental test-bed consisting of 16 sensor nodes deployed indoors for the purpose of testing a positioning system based on WiFi and Bluetooth technologies. Therefore, the chosen parameter setting were specifically tailored to the system. Still, the described approaches of filtering and data aggregation can fit to a diversity of WSN applications by simply alternating component combinations (filtering) or fine tuning of parameters (aggregation). For example, an environmental monitoring application can tolerate long aggregation periods, radio echo profiling can use only whitelisting the deployed sensors but an assisted/ambient living application may prefer pure blacklisting, since whitelisting requires human participation.

The presented evaluation and results have relevance beyond the scope of wireless test-beds. We are confident that filtering and aggregation strategies can also help real-world deployments to scale better and to make better use of the limited radio resources. We are aware that both mechanisms have a downside, e.g., wrongly identifying a device as always-on in filtering or loosing measurements details in aggregation, but we believe that a careful parametrisation can eliminate the effects. In addition, compression techniques could further bring the size of the transferred data down.

References

1. Greenorbs test-bed. `http://greenorbs.org`. Accessed: 27/01/2012.
2. Gumstix overo. `https://www.gumstix.com/store/product_info.php?products_id=227`. Accessed: 27/01/2012.
3. Honk kong university, internet and mobile computing laboratory test-bed. `http://www.comp.polyu.edu.hk/en/research/centres_labs/internet_and_mobile_computing_laboratory/index.phpm`. Accessed: 27/01/2012.
4. IEEE 802.11-2007, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. `http://standards.ieee.org/about/get/802/802.11.html`, 2007.
5. Z. Chen and K.G. Shin. Opag: Opportunistic data aggregation in wireless sensor networks. In *Real-Time Systems Symposium, 2008*, pages 345 –354, 2008.
6. D.C. Dimitrova, I.. Alyafawi, and T. Braun. Experimental comparison of bluetooth and wifi signal propagation for indoor localisation. In *10th International Conference on Wired/Wireless Internet Communications*. to be published in LNCS, 2012.
7. K. Dolfus and T. Braun. An evaluation of compression schemes for wireless networks. pages 1–6. International Congress on Ultra Modern Telecommunications and Control Systems, 2010.
8. L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Proc. of Distributed Computing Systems Workshops*, pages 575 – 578, 2002.
9. V. Kumar, J. McCarville-Schueths, and S. Madria. A test-bed for secure hierarchical data aggregation in wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pages 762 –764, nov. 2010.
10. R.N. Murty, G. Mainland, I. Rose, A.R. Chowdhury, A. Gosain, J. Bers, and M. Welsh. Citysense: An urban-scale wireless sensor network and testbed. In *Proc. of IEEE Technologies for Homeland Security*, pages 583 –588, 2008.
11. T. Staub, S. Morgenthaler, D. Balsiger, P.K. Goode, and T. Braun. Adam: Administration and deployment of adhoc mesh networks. 3rd IEEE Workshop on Hot Topics in Mesh Networking (IEEE HotMESH 2011), 2011.
12. Z. Taghikhaki, N. Meratnia, and P.J.M. Havinga. Energy-efficient trust-based aggregation in wireless sensor networks. In *IEEE INFOCOM: Workshops*, pages 584 –589, 2011.