
Reliable localised event detection in a wireless distributed radio telescope

Suhail Yousaf, Rena Bakhshi and Maarten van Steen

Department of Computer Science,
The VU University Amsterdam,
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands

and

The Network Institute, The Netherlands

E-mail: s.yousaf@vu.nl

E-mail: rbakhshi@few.vu.nl

E-mail: m.r.van.steen@cs.vu.nl

AQ1

Abstract: We consider a large wireless network constituting a radio telescope. Each of the anticipated 3000 nodes is triggered to collect data for further analysis at a rate of more than 200 Hz, mostly caused by noisy environmental sources. However, relevant cosmic rays occur only a few times a day. As every trigger has an associated 12.5 KB of data, and considering the size of the telescope in number of nodes and covered area, centralised processing is not an option. We propose a fully decentralised event detection algorithm based on collaborative local data analysis, effectively filtering out only those triggers that need further (centralised) processing. As we show through performance evaluations, the crux in the design is finding the right balance between accuracy and efficient use of resources such as the communication bandwidth in the unreliable communication environment.

Keywords: distributed event detection; large-scale distributed systems; in-network data processing.

Reference to this paper should be made as follows: Yousaf, S., Bakhshi, R. and van Steen, M. (xxxx) 'Reliable localised event detection in a wireless distributed radio telescope', *Int. J. Sensor Networks*, Vol. x, No. x, pp.xxx-xxx.

Biographical notes: Suhail Yousaf received his MSc in Parallel and Distributed Computer Systems from VU University Amsterdam (2009). Currently he is a PhD student in the Computer Systems group at VU University Amsterdam. He is developing in-network algorithms for data processing in wireless sensor networks with extreme conditions: large-scale, constrained resources, high accuracy and efficiency demands.

Rena Bakhshi received her MSc in Computer Science from Royal Institute of Technology in 2006, and PhD from VU University Amsterdam in 2011. She currently works as a postdoctoral fellow in Computer Science Department, at VU University Amsterdam. Her research focuses on modelling and analysis of large-scale stochastic systems.

Maarten van Steen is Full Professor at VU University Amsterdam where he currently heads the Department of Computer Science. He concentrates on large-scale distributed systems, including very large wireless sensor-based networks. His research encompasses low-level systems work including the development of novel MAC-layer protocols, in-network algorithms for data aggregation and epidemic-based data dissemination, as well as frameworks for capturing properties in terms of complex systems. He is author of several textbooks, including 'Distributed Systems', co-authored with Andrew Tanenbaum, and a recent introductory textbook on *Graph Theory and Complex Networks*.

1 Introduction

In a geospatial sensor network, a node is responsible for gathering location-sensitive data. In many cases, as these networks grow in diameter (measured in meters), as well as in their number of members, we often see that communication paths can be mainly realised only through wireless channels. A typical example that we consider in this paper is a large-scale radio telescope consisting of a few thousand nodes spread over an area of around 3000 km². An important consequence of this growth is that the processing of data needs to be localised, as realising communication paths to centralised, specialised nodes becomes more difficult.

In this paper we consider a specific sensor network that we believe is characteristic for many other Wireless Sensor Networks (WSNs). First, each node is responsible for detecting specific, often rare, events, in our case detecting cosmic rays. Event detection requires sampling data. However, a node may be operating in an extremely noisy environment meaning that there can be many false detections leading to an explosion of sampled data of which most will turn out to be useless. It is crucial that such data is kept local and preferably not sent to any other node, let alone a centralised base station or the equivalent thereof.

Second, to distinguish between relevant and false events, a node requires information from its immediate (geographical) neighbours: an event is relevant only if some of the neighbours have detected it as well. This means that data from multiple nodes will need to be combined before coming to conclusions on relevancy of an event. In some sensor networks, the base station simply gathered the data from all nodes, but as we argued, this approach must be abandoned when networks grow.

We concentrate on the problem of distributed event detection in a large WSN in which neighbouring nodes collaborate to filter out relevant events before communicating associated data to one or several central nodes. Each node has not only a limited energy budget, but also other resources are scarce such as memory and processing capacity. In addition, as we are dealing with wireless networks; communication links are unreliable.

For our specific application, cosmic-ray detection, relevant events are so rare that we essentially cannot afford to lose any of them. In other words, much effort should be put into keeping the fraction of false negatives close to zero. On the other hand, false positives should also be minimised, but for a different reason, namely that of minimising resource consumption and thus maximising efficiency. However, optimising for efficiency becomes more relevant when realising that events occur at a rate of approximately 200 Hz, yet that less than 1% is actually relevant. In other words, a huge data-processing effort is required for successfully detecting cosmic rays.

We make the following contributions. First, we present a distributed, in-network event detection

algorithm based on collaborative local data analysis that reduces resource consumption in large-scale geospatial sensor networks. Second, we investigate the application-level resource usage such as the communication bandwidth for a certain level of performance in unreliable communication environment. This is the first paper to our knowledge to explore the possibility of applying collaborative local data analysis in large-scale geospatial WSNs to detect ultra-high energy cosmic rays.

The rest of this paper is organised as follows. In Section 2, we give an overview of the related work in the field of the monitoring applications. Section 3 describes our distributed event detection algorithm designed for large-scale sensor networks. In Section 4, we discuss the methodology used to analyse the performance of the our proposed approach. In Section 5, we present the results based on experiments for investigating bandwidth requirements of the application. In Section 6, we present and discuss a rigorous performance evaluation of our proposed algorithm. Finally, Section 7 concludes the paper.

2 Related work

The common model for event detection in a WSN is that each node simply relays all of its locally generated data to the base station without local processing. The data is processed for event detection only at the base station (Gehrke and Madden, 2004). This model works well for small-scale networks, a small amount of data per event, and lower frequencies of events per node. It is unacceptably inefficient for large-scale networks, as it may involve considerable bandwidth consumption as multiple communication hops need to be taken.

Another model for event detection involves in-network processing. To this end, processing is done by the nodes to compute events of interest against criteria known to the node. This may significantly reduce the amount of communication and, hence, the energy consumed. The most commonly used techniques for in-network processing in WSNs are:

- processing along a routing path to the base station (Madden et al., 2002)
- processing at regional head nodes (Martincic and Schwiebert, 2006; Werner-Allen et al., 2006)
- initiating in-network consensus (Werner-Allen et al., 2005)
- deciding locally at a node based on information from its neighbours (Wittenburg et al., 2010).

However, all the proposed schemes have different shortcomings for our application:

- there is no acknowledgement of the event detection by a node in Madden et al. (2002)

- the scheme in Martincic and Schwiebert (2006) produces false negatives in case an event occurs on the border of two or more cells
- a low event frequency is assumed in Wittenburg et al. (2010)
- the schemes in Werner-Allen et al. (2006) and Werner-Allen et al. (2005) assume network-wide events, and thus, are not scalable for large geographical areas.

There are two classes of work done on cosmic-ray detection. A direct cosmic-ray detection method (Müller et al., 2007; CREAM, n.d.; AMS, n.d.) needs a high-altitude balloon or a satellite/space mission, and detects only low-energy particles. To detect the much rarer highest energy particles, an indirect cosmic-ray detection method is needed, such as Aramo (2005), Huege (2010) and Blümer et al. (2009). However, these systems use a wired backbone (fibre optic) for communication and, therefore, suffer seriously from geographical scalability issues.

3 Distributed event detection

3.1 System model

In the context of cosmic-ray detection, we consider a large field covered by a large collection of stations, each equipped with a wireless sensor. They sense radio signals and communicate with neighbouring stations in the field through a low-power wireless medium. Each station has limited processing capabilities, energy budget and a storage capacity in the order of a few hundred megabytes. The clocks of stations are globally synchronised via integrated GPS receivers. (The accuracy is within 1–2 nanoseconds through special devices (Kelley and the Pierre Auger Collaboration, 2011)). Each station relays its data to a base station called the Central Radio Station (CRS) for further analysis.

The stations are stationary and location-aware. We assume direct communication only between stations within a certain distance (*geographical neighbours*). Each station captures radio signals with a certain strength into a so-called *N1 trigger*, which may indicate the occurrence of a cosmic-ray. In fact, the N1 trigger is equivalent to what is called the ‘level 2’ trigger in Kelley (2012). Each trigger is timestamped at nanosecond accuracy. The timestamp is a pair of `seconds` (date and time into the UNIX epoch, in UTC) and `nanoseconds`. For each trigger, in addition to the timestamp, a digitised portion of the signal of 12.5 kilo bytes is also buffered at the station. This data along with the timestamp is sent to the CRS upon positive decision through a data analysis procedure; otherwise both the timestamp and buffered data are ignored.

The triggers of two geographically neighbouring stations *coincide* if their timestamp difference ΔT is less than T_c , the light-travel time in a straight line from one station to the other station. An N1 trigger is promoted to an *N2 trigger* if it is coincident with an N1 trigger of a geographical neighbour. An N1 trigger at a station is promoted to an *N3 trigger* in two cases:

- the N1 trigger at a station is coincident with N1 triggers of at least two other geographical neighbours
- the N1 trigger at a station is coincident with an N3 trigger of any of its geographical neighbours.

Note that we also call an N3 trigger an *event of interest*.

The direction of the signal that caused the N3 trigger is reconstructed using timestamps and geographical positions of the stations that took part in the coincidence. The direction reconstruction uses what is known as *plan wave fit*. The reconstructed direction is used as a tuple of *zenith* and *azimuth* angles. The reconstructed direction helps in deciding whether to discard an L3 trigger caused by some man-made noise source.

3.2 The algorithm

Whenever an N1 trigger occurs at a station, the station stores the N1 trigger locally and informs all of its neighbours by sending them the timestamp of its N1 trigger. Furthermore, when a station receives N1 triggers from its neighbours, it looks for a coincidence of the received triggers with its local ones. A station promotes its N1 trigger to an N3 trigger if its N1 trigger has coincidence with N1 triggers of at least two neighbours. To cover stations on the boundary of an event region with only one geographical neighbour in the event region, a station is not only required to broadcast its N1 triggers, but also to advertise its N3 triggers. This message helps a station promote its N1 trigger to an N3 trigger if its N1 trigger coincides with an N3 trigger contained in the advertisement message. To reduce bandwidth consumption, the algorithm uses periodic broadcast messages, which we call an *N1 bundle*, by simply grouping together the N1 triggers. Similar to the N1 bundle formation, the local N3 triggers are bundled as an *advertisement bundle*.

Figure 1 shows the pseudocode of our algorithm. When an N1 trigger occurs at a station s , it adds the trigger to its local cache. The trigger is also added to a local N1 bundle that will be broadcast to geographical neighbours of the station. The algorithm executes two threads: an active and a passive one. The active thread is executed periodically. It broadcasts N1 bundles and advertisement bundles of a station to the geographical neighbours of the station. The passive thread listens to incoming messages. Upon receipt of an N1 bundle or an advertisement bundle from a geographical neighbour, the thread looks for a coincidence of each trigger

in the bundle with the local N1 triggers. Whenever an N1 trigger is promoted to an N3 trigger (see, $N1(s) \rightarrow N3(s)$ in Figure 1), the station will execute the operation $process(N3(s))$ if further processing of the trigger is required (e.g., applying any domain-specific filter to the N3 trigger).

Figure 1 Pseudocode for our algorithm

```

/** On Local N1 Trigger */    /** Passive thread */
// Runs when a local N1 trigger // Runs when receiving an
// occurs at station s       // N1Bundle or AdvertBundle
localCache.add(N1(s))        receive < N1Bundle(q), q >
N1Bundle.add(N1(s))          OR
                              receive < advertBundle(q), q >

/** Active thread */
// Runs every T seconds
for all q ∈ Neighs do
  send < N1Bundle(s), q >
for all q ∈ Neighs do
  send < AdvertBundle(s), q >

/** On Remove Trigger */
// Runs when a local N1 trigger is
// marked for removal under the
// cache eviction policy
if NOT isDecided(N1(s)) then
  apply user defined criteria

for all entry ∈ Bundle do
  if localCache.coincidence(entry) then
    N1(s) → N3(s)
    process(N3(s))
  if entry ∈ N1Bundle then
    advertBundle.add(entry)
    localCache.remove(N1(s))

```

If a coincidence has been found between the local N1 trigger and N1 triggers of the geographical neighbours, then these N1 triggers are added to the advertisement bundle. On the other hand, if station s promotes its N1 trigger to an N3 trigger because of coincidence with an N3 trigger of any of its geographical neighbours, the promoted N3 trigger is not broadcast to the neighbours. The reason for this is that station s has only one neighbour in the event region that had already promoted its corresponding N1 trigger to N3. For station s , broadcasting the advertisement in this case is useless.

The local triggers are stored in a buffer with limited capacity. Depending on local processing capabilities and available memory, we need to take into account that the buffer may become full. As a consequence, a *cache eviction* strategy is required. For example, our algorithm can employ a simple strategy that discards newly arriving triggers while old triggers have not yet been removed. Alternatively, we could also decide to remove the oldest trigger, or choose one randomly. These strategies are rather straightforward, and are application oblivious. A more optimal cache eviction strategy may require further insight into the semantics of triggers. In this paper we take a simple approach and remove the oldest triggers when the buffer becomes full, except the ones that are currently being examined.

4 Experimental setup

To evaluate the performance of our algorithm in the presence of communication failures, we have conducted a set of experiments using the OMNET++ simulation environment (Varga, n.d.). This section discusses the methodology used to evaluate the performance of our algorithm.

4.1 Network specification

In the simulated network, the stations are placed in a triangular grid topology with an average inter-node distance of 150 m. Note that the triangular grid topology was the requirement of the application. To simulate the behaviour of a wireless ad-hoc network, we make a distinction between the so-called application layer and the system layer. The *application layer* consists of our basic algorithm with configuration parameters. The *system layer* is an application-independent wireless network. In essence, our solution consists of two sets of algorithms. The first contains the core of our solution and its algorithms essentially assume that the system layer operates flawlessly and with infinite resources. The second set contains algorithms that compensate the shortcomings of the system layer: lossy links, faulty nodes, and limited resources. The core algorithms will always need to be executed; the compensation algorithms are executed only because the system layer is far from being perfect.

Executing the core algorithms will demand a certain capacity from the system layer in terms of bandwidth usage, memory usage, energy consumption, etc. Because of failures in the system layer, our compensation algorithms will also need to be executed, requiring further capacity. We are interested to know how much capacity the application layer requires from the system layer in the presence of faulty links and nodes. This will help in choosing a suitable system layer technology for further experimentation and eventual deployment of the system.

To analyse the capacity required by the application layer, we take a simple approach by initially assuming that only links can fail, in particular with a probability p . In other words, the system layer is based on a probabilistic link-failure model and is denoted as SL- p . We furthermore consider only the use of bandwidth, as this is most likely our scarcest resource. We vary p to see

- to what extent additional bandwidth is needed for the compensation algorithms
- what is the impact on performance.

In this way, we aim at obtaining an upper bound for the bandwidth capacity that the system layer should provide.

Essentially the experiments were performed in two phases. The first phase was aimed at exploring the performance of the algorithm over the entire spectrum of link-loss probability p . At this phase, we were especially interested in gaining an insight in the bandwidth requirement imposed by the application layer in the presence of lossy communication. These experiments played a crucial role in deciding about the system layer technology, required to evaluate the algorithm, considering real radio models with bandwidth constraints and more detailed link-loss

patterns in the network. In the second phase, experiments were conducted using a real-world system layer specification that was chosen based on lessons learned in the first phase. The aim of these experiments was to rigorously evaluate the performance of the algorithm.

4.2 N1 trigger traces

We emphasise that our exploration is based on real data. Traces of N1 triggers are used that were collected from a small-scale, real-world testbed for cosmic-ray detection (Kelley, 2012). The testbed consists of 24 stations and uses wired infrastructure for communication between stations and the central unit CRS. The data analysis procedure in the testbed is centralised: every station sends its N1 triggers to the CRS for noise filtering and further analysis. However, the occurrence of N1 triggers is independent of the data analysis procedure itself. Therefore, the N1 triggers generated in the testbed could still be used for our algorithm based on collaborative local data analysis, and executed on top of a wireless system layer.

To establish a baseline for evaluating the performance of our algorithm, we ran the traces through an ideal system and observed which nodes actually promoted their N1 triggers to N3 triggers eventually. Next, the same traces were run through our own algorithm, executed in the simulated environment. In this way we were able to determine that our system is functioning correctly.

4.3 Performance metrics

Ideally, if an N1 trigger at a station is not promoted to an N3 trigger, it is considered to be noise and must be discarded by the station. However, in the presence of a system layer with unreliable communication, there is always a chance to discard an N1 or N2 trigger that is actually an N3 trigger, but due to communication failures could not be promoted to N3. This situation will result in a *false negative* (f^-). Moreover, a *false positive* (f^+) is produced when a trigger that is not an N3 trigger is chosen to be reported to the CRS.

We consider the performance of our algorithm in terms of its accuracy and efficiency. The *accuracy* is measured by the number of f^- produced. The *efficiency* is determined by the number of f^+ .

4.4 System parameters

There are a number of system parameters that influence the performance of our algorithm. This section briefly explains those parameters.

Bundling: Each station *bundles* its triggers before broadcasting to its neighbours. Bundling helps reduce the message size by eliminating redundant information within the message. For further details on how does bundling work, please refer to Yousaf et al. (2012).

Message compression: The message size can be reduced by using lossless compression techniques. This means message compression helps in reducing bandwidth consumption, nevertheless, at the cost of increased local computation and memory usage.

Broadcast frequency: Broadcast frequency is defined as the number of unique messages broadcast by a station to its first-hop neighbours, per time unit. The N1 triggers occurred at a station during a time interval $\Delta t = t_i - t_{i-1}$ are bundled in a single message. The length of the time interval Δt is a system wide parameter. In other words,

$$\text{broadcast frequency} = \Delta t^{-1}. \quad (1)$$

Message redundancy: A considerable number of messages are lost due to unreliable communication. This can affect the performance of the algorithm. To compensate potential message loss, message redundancy can be considered where a message is broadcast more than once by the source station. In principle, redundancy of messages should increase the chances of message delivery in an unreliable communication environment.

Handling undecided triggers: In case of undecided triggers remaining at a station, there are three options. First, discard all undecided triggers. Second, report all undecided triggers (*i.e.*, N1 and N2) to the CRS. Third, report only undecided N2 triggers to the CRS. We analyse the performance of the algorithm for each of the three cases.

5 Phase I: Bandwidth requirement analysis

This section presents results obtained from the first phase of our experiments aimed at investigating the bandwidth requirement imposed by the application layer on the system layer. The system layer SL-p is used. The SL-p does not take into account the size of the message. Since broadcast frequency dictates message size, it is insignificant to explore a range of broadcast frequencies. Therefore, we keep the broadcast frequency fixed. For simplicity, it is set to 1Hz. In this phase we do not consider message compression for the sake of determining an upper bound on the bandwidth requirement. We consider message redundancy as well as the handling of undecided triggers as both of these parameters are crucial in bandwidth requirement specification.

5.1 Results

We first validate our collaborative, local data-analysis approach. To that end, we executed our algorithm in an environment with perfect communication. Each station indeed observed the same N3 triggers as the ones obtained through the existing centralised approach,

described earlier. This means that our algorithm behaves correctly.

First we analyse the performance of the algorithm assuming perfect communication among the stations. The assumption is made for two reasons. First, to demonstrate the potential of local data analysis to filter out relevant data. Second, the obtained performance will be used as reference for comparison. Table 1 depicts the filtering capability of the algorithm. Considering the whole network of stations as a single entity, we see that the network observes a huge number of N1 triggers. Note that these N1 triggers have been observed over a period of 100 s. The algorithm processes these triggers locally. Only one third of the triggers is promoted to an N3 trigger. By definition, an N3 trigger is called an *event of interest*. The N1 triggers at a station that were not promoted to N3 triggers are discarded locally. Each station furthermore applies a so-called direction reconstruction (DR) filter (Huege, 2010) to its N3 triggers. The filter discards those N3 triggers whose corresponding directions point to the horizon. The *zenith* angle in the range 90 ± 5 is considered as horizon. We see that the number of relevant triggers to be sent to the CRS are further reduced by applying the *DR* filter. So, under the assumption of perfect communication, the ratio $\left(\frac{DR(N3)}{N1}\right)$ indicates that the algorithm is able to discard up to 83% of the triggers locally.

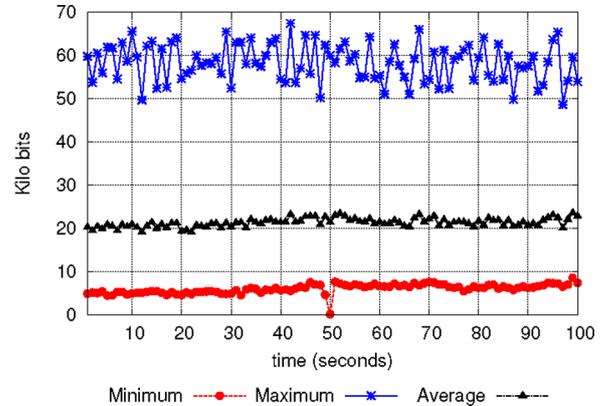
Table 1 Filtering capability of the algorithm

<i>Triggers type</i>	<i>No. triggers</i>
N1	583,455
N2	323,099
N3	218,202
DR(N3)	93,932

The algorithm demands a certain capacity from the *system layer* to process the triggers occurred at stations. The most crucial is the bandwidth required by the algorithm. To that end, we measure the bandwidth (imposed by each station on the system layer) in consecutive time windows each of length T_w . We compute the maximum, minimum, and average bandwidth required per station during T_w . For simplicity, we assume $T_w = 1$ s. Figure 2 shows the temporal dynamics of these metrics. We see that the maximum is far away from the corresponding average. In general, there are two possibilities for the large gap between maximum and average. First, there may be a specific station continuously triggering with high rate and generating a relatively higher amount of data. Consequently, each time the maximum for T_w is contributed by this particular station. A second possibility is that different stations during different T_w produce burst data that pushes the maximum away from average. In our specific trace, we noticed that there are a few neighbouring stations that trigger with high rate and push the maximum upward. In principle,

the system layer should be able to absorb the imposed maximum bandwidth irrespective of the underlying cause.

Figure 2 A temporal view of the bandwidth production by the algorithm (see online version for colours)



Next we analyse the performance of the algorithm assuming that communication links at the system layer may fail with a probability p . Due to link failures a station may not be able to receive enough information from its neighbours to decide about its local N1 triggers. In this situation, discarding those N1 triggers that were not promoted to N3 triggers due to lack of information will give rise to false negatives. Our aim is to keep the number of false negatives low. To this end, we use our compensation algorithms that basically produce message redundancy in the network; thus increasing the chances of message delivery. To establish a basis for comparison, a fixed rebroadcast is used where each message is broadcast twice. It is thereby assumed that a link is temporarily down with probability $p = 0.5$ when the message travelled through it.

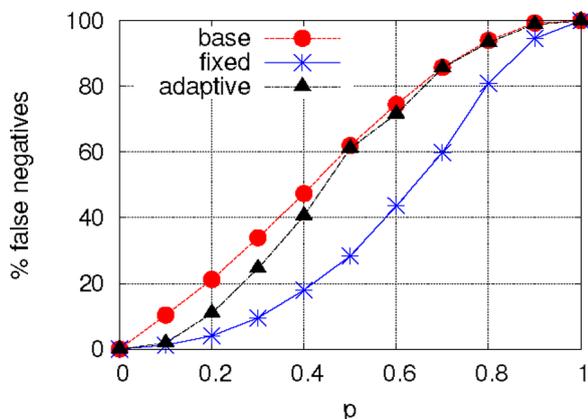
In addition to fixed rebroadcast, adaptive rebroadcast is used where each station maintains a Link Quality Estimator (LQE) (Woo et al., 2003). A message is rebroadcast only if the weakest link quality is above 50% and below 100%. The links with quality below 50% are considered dead, therefore, rebroadcast is abandoned.

We examine the impact of compensation algorithms on performance. More specifically, three different cases are considered. First, the core algorithm is executed without any compensation algorithm; the so-called base case. Second, the core algorithm is executed with fixed rebroadcast. Third, the core algorithm is executed with adaptive rebroadcast. All the three cases are repeated with various link loss probabilities. We are interested to see the extent to which the compensation algorithms compensate the link failures by reducing the number of false negatives.

Figure 3 shows a comparison of the three cases for a range of message-loss probabilities. For $p = 0$ all three cases have no false negatives because of no messages are lost. For $p = 1$ all three cases have 100% false negatives for the obvious reason that all messages are

lost by the system layer and none of the stations is able to compute N3 triggers. The cases with message redundancy produce fewer number of false negatives than the base case. This shows the effectiveness of the compensation mechanisms in improving the accuracy of the core algorithm. The adaptive case performs better than the base case only when $p < 0.5$. This is because of the way the adaptive rebroadcast works. For $p \geq 0.5$ the link qualities computed through the LQE are mostly below 50%. Since rebroadcasts are abandoned for link quality below 50%, the adaptive case behaves similar to the base case for $p \geq 0.5$. The fixed rebroadcast is expected to produce higher redundancy than the adaptive rebroadcast. The result is that it outperforms the adaptive case by producing fewer false negatives.

Figure 3 The effect of compensation algorithms on false negatives (see online version for colours)

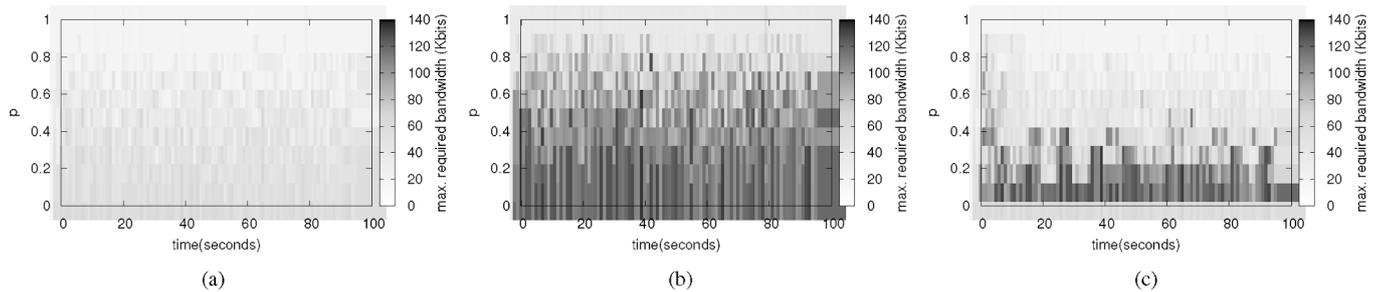
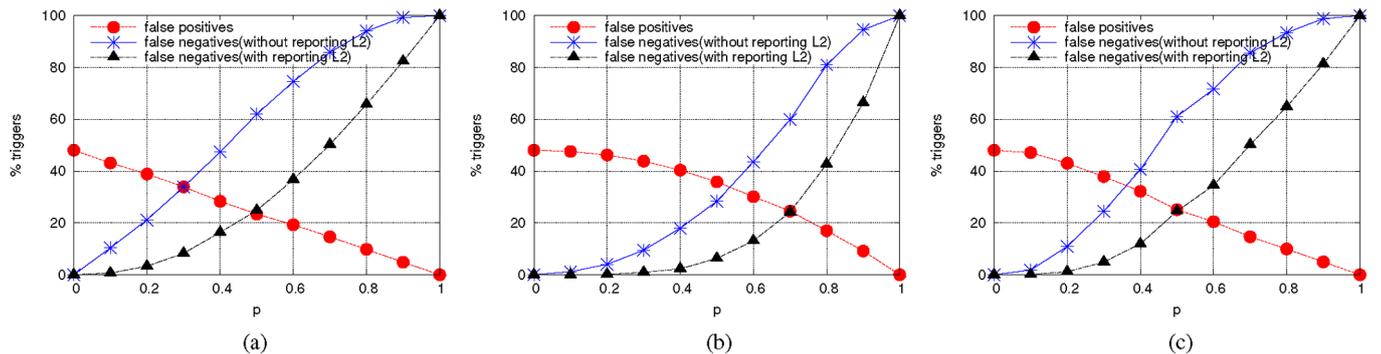


We examine the worst-case bandwidth requirements of our compensation algorithms by computing the maximum bandwidth required per station per time window T_w . For simplicity we assume $T_w = 1$ s. Figure 4 depicts the effect of compensation algorithms on the worst-case bandwidth requirements. For $p = 0$ the maximum bandwidth requirements for the adaptive case is the same as the base case. The reason is that due to no message loss the adaptive case does not rebroadcast messages and produces exactly the same maximum bandwidth as the base case. For $p = 0$, the fixed case produces bandwidth that is twice as much as the base case because every message is broadcast twice and communication is error-free. The reason is that the fixed case broadcasts every message twice. Due to no loss of messages the produced bandwidth is exactly double of the corresponding base case. A similar behaviour can be observed for $p = 1$. Every station is unable to compute N3 triggers due to unavailability of information from its neighbours. So a station broadcasts only its local N1 triggers. There are no rebroadcasts in the adaptive case because for $p = 1$ the link quality remains zero and rebroadcast is abandoned. The maximum bandwidth requirement of the base case is less than both the fixed case and the adaptive case. However, the base case produces comparatively more false negatives. On the other hand, the fixed case imposes the highest

bandwidth requirement but produces the least number of false negatives. The adaptive case tries to reduce bandwidth production by selectively rebroadcasting. The adaptive case reduces bandwidth compared to the fixed case but at the cost of producing a higher number of false negatives than the fixed case. We see that more accurate event detection requires additional bandwidth. The system layer is required to meet the worst case bandwidth requirement in order to keep the number of false negatives within a certain limit.

Another approach to minimise false negatives is the way the algorithm handles undecided triggers local to a station. We consider a system layer with faulty communication links. Therefore, it is likely that a station may not be able to receive sufficient information from its neighbours. Consequently, some of the local triggers (including N1 and N2) at the station may remain undecided. There are three different options to handle these undecided triggers. First, discarding all the undecided triggers, a station may falsely discard many triggers. This will lead to a high number of false negatives. Second, reporting all undecided triggers as false positives will exclude the possibility of false negatives but this will push the number of false positives to a maximum. In fact, this option is equivalent to reporting everything to the CRS. Third, the algorithm reports only undecided N2 triggers to the CRS. There are two assumptions underlying this option. First, an N2 trigger is an N3 trigger but due to communication failures it was not promoted to an N3 trigger. In case the assumption is correct, false negatives will reduce without an increase in false positives. Second, it is assumed that an undecided N1 trigger is random noise and must be discarded. Again, if the assumption is correct then neither false positives nor false negatives will increase. Otherwise a node may falsely discard the N1 trigger.

Figure 5 depicts the effect of reporting undecided N2 triggers on the performance of our three cases: base, fixed and adaptive for various link loss probabilities. For $p = 0$, all cases have the same number of false positives. The reason is that for $p = 0$, the communication is reliable and all possible N3 triggers are successfully computed. The N2 triggers reported in this case are definitely noise. For $p = 1$, there are no false positives. By definition, a false positive must be an N2 trigger. Since, for $p = 1$, all messages are lost no station is able to compute an N2 trigger. Therefore, the number of false positives drops to zero. On the other hand, we see that the number of false negatives is maximal; all undecided N1 triggers are falsely discarded. The decline in false positives with increase in p is due to the fact that fewer N2 triggers are computed in a more lossy environment. For that reason, we see a sharp decline in false positives for the base case. In general, we see that by reporting undecided N2 triggers the number of false negatives decrease. Moreover, the rate of decrease in false negatives is higher in cases where the core algorithm is executed in combination with some

Figure 4 The impact of compensation algorithms on maximum bandwidth production: (a) the core algorithm without rebroadcasts; (b) the fixed rebroadcasts and (c) the adaptive rebroadcasts**Figure 5** The effect of reporting undecided N2 triggers to the CRS: (a) the core algorithm without rebroadcasts; (b) the fixed rebroadcasts and (c) the adaptive rebroadcasts (see online version for colours)

compensation algorithm. However, the higher decrease in false negatives is at the cost of higher number of false positives in the corresponding cases.

5.2 Lessons learned

Our distributed event detection algorithm does not produce false negatives or false positives in an ideal communication environment. However, due to unreliable wireless communication the algorithm suffers from false negatives. Since high-energy cosmic rays are extremely rare, false negatives are unacceptable. To reduce false negatives we evaluated two approaches: message redundancy and reporting partially aggregated data (N2 triggers). However, message redundancy requires additional bandwidth. Similarly, reporting N2 triggers causes false positives which are also resource consuming.

We notice that under ideal circumstances, the bandwidth requirements exceed what many networks solutions (e.g., Zigbee) can currently provide. We have not yet considered saving energy by going into duty-cycle mode or increasing the neighbourhood to increase robustness at the cost of sharing the available bandwidth among more neighbours. In short, we have to be careful on the selection of system layer technology used for wireless communication. The next section discusses the system layer technology selection and the second phase of our experiments conducted to analyse the performance of the algorithm executed on top of the selected system layer technology specification.

6 Phase II: Performance analysis

This section covers the second phase of our experiments. First, we present a discussion on system layer technology selection. Next, the system configuration parameters are discussed. Finally, we present and discuss the results obtained from the experiments in this phase.

6.1 Technology selection

There are three major constraints to be considered while establishing our network of stations for cosmic-ray detection. These include the interstation distance, energy budget, and communication bandwidth. There are two considerations for the interstation distance. In the first phase a medium-scale deployment will consist of 160 stations with interstation distance of 150 m. In a subsequent large-scale deployment, with thousands of stations, the interstation distance will be extended to 1500 m. Every station is powered by an energy-harvesting device, limiting the station energy budget for communication to 3 Watts. On the other hand, the high trigger rate per station demands relatively high bandwidth.

We opt for a communication technology that, in principal, is commercially available. There are not many options that meet the constraints we are dealing with. ZigBee-Pro offers an outdoor communication range up to 1500 m, but only 250 kbps of a bandwidth. The 802.11x offer bandwidth upto 65 Mbps but the outdoor communication range is restricted to 300 m.

We chose the ZigBee-Pro specification which offers the transmission range that is required for our large-scale deployment. It offers the required transmission range with a transmit power of 100mW which is within our energy budget. It turned out that a single ZigBee-Pro module per station is insufficient to handle the bandwidth produced by the application layer. Thus, multiple modules per station were assumed. However, the choice of the number of modules per station should also consider the energy budget available for communication.

For our study, the stations in the simulated network are deployed on a triangular grid with an interstation distance of 150 m. It is assumed that the energy consumption is below the available budget per station. We employ the standard ZigBee-Pro configuration; the radio propagation model is tuned such that a station can communicate only with its geographical neighbours.

A system layer should offer enough bandwidth required by the application layer. The building block of the system layer for our experiments is the ZigBee/IEEE-802.15.4 module that offers a bandwidth of 250 kbps per channel out of the available 16 channels. We consider three reference system layers based on their bandwidth capacity, namely

- *SL-250 kbps*
- *SL-750 kbps*
- *SL-2 Mbps* that mimic one, three, and eight ZigBee modules per station, respectively.

6.2 System parameters

The following system parameters were considered to evaluate the behaviour of the algorithm.

Message compression: We study the effect of lossless message compression on bandwidth usage. For this purpose, every station uses the lossless compression library *zlib-1.2.5* to compress its outgoing message and decompress the received messages from its neighbours.

Broadcast frequency: The broadcast frequency plays a crucial role in this phase of the experiments. The message size is approximated by the broadcast frequency. On the other hand, the system layers that are used in this phase are sensitive to message size. For example, longer messages have higher chances of suffering from bit-errors. Therefore, it is important to explore the performance of the algorithm for a range of broadcast frequencies. We consider broadcast frequencies in the close interval $[1, 40]$ of natural numbers.

Message redundancy: The selected system layers are based on lossy communication. This means that some messages are lost due to collisions and various other reasons. To compensate the message loss, we introduce message redundancy at the application layers. The message redundancy is based on fixed rebroadcasting.

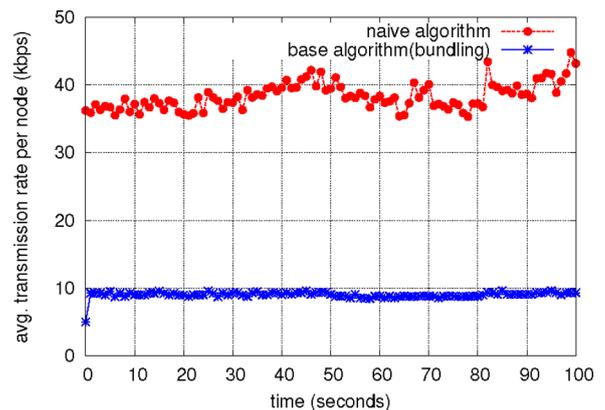
More specifically, the application layer at a station hands over every message to the system layer two times for broadcasting to its first hop neighbours.

Handling undecided triggers: The selected system layers for this phase of the experiments use real-world specifications. Therefore, it is important to thoroughly analyse the performance of the algorithm while considering the different options, discussed in Section 4.4, for handling undecided triggers.

6.3 Results

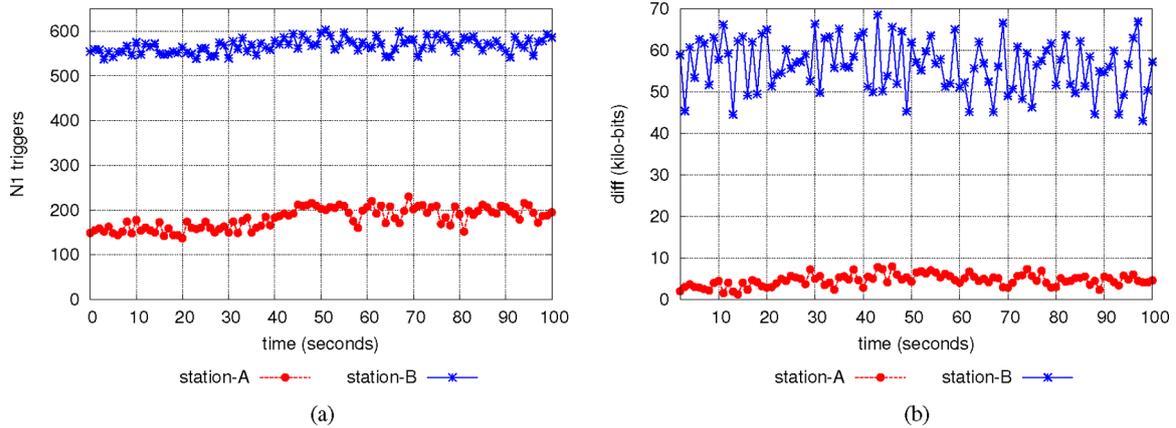
The performance analysis begins by evaluating the effectiveness of *bundling*. As discussed in Section 4.4, bundling can help in reducing bandwidth consumption. We compared the bandwidth consumption of our base algorithm (which uses bundling) with a naive algorithm where each trigger is broadcast immediately after its occurrence. A comparison of the two algorithms is shown in Figure 6; bundling reduces bandwidth consumption by approximately a factor of 4.

Figure 6 Effect of bundling on bandwidth consumption (see online version for colours)

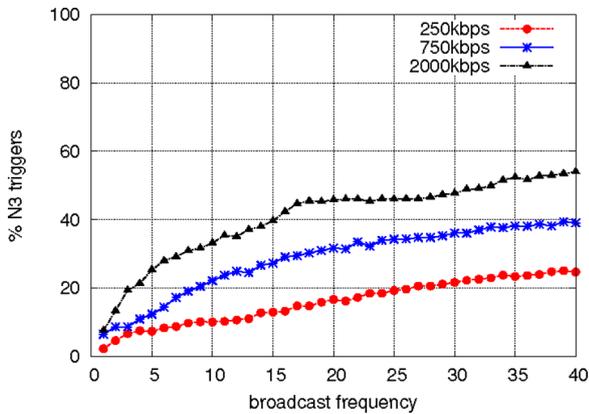


We evaluate the effect of lossless message compression on bandwidth production of our algorithm. As shown in Figure 7(a), we consider two stations *A* and *B* with a low and high N1 trigger rate respectively. Figure 7(b) shows the reduction in bandwidth consumption due to compression; compression is more effective for station *B* with a higher rate of N1 triggers. Since message compression is computationally expensive, it is not always desirable to apply it when there is no significant gain. However, it is possible to make the decision of compressing a message dependent on the local N1 trigger rate. In this case message compression is applied only when a significant gain is expected; otherwise, the message is broadcast without compression.

Next we consider the accuracy of our algorithm. Ideally, the algorithm is expected to be 100% accurate which means that it should detect all N3 triggers. However, due to communication failures the algorithm may not be able to maintain the ideal accuracy. The algorithm is executed on top of each of our

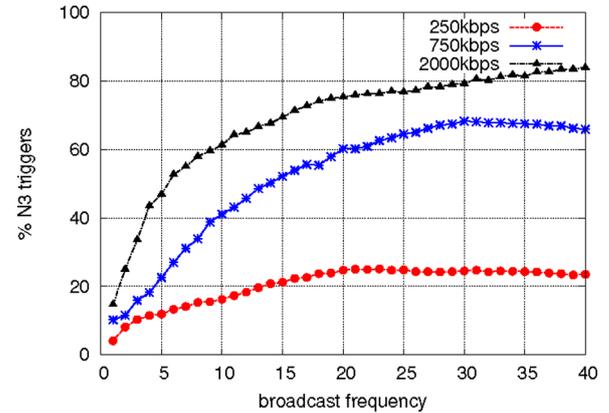
Figure 7 Effect of message compression on bandwidth consumption: (a) N1 trigger rate and (b) diff. due to compression (see online version for colours)

three reference system layers. To see the effect of broadcast frequency on accuracy of the algorithm, different broadcast frequencies are used for the same system layer. A comparison of the accuracy level for various broadcast frequencies is shown in Figure 8. The level of accuracy also improves with the increase in available bandwidth at the system layer, which allows for the higher broadcast frequencies. Moreover, the SL-750 kbps system is a moderate choice among our three reference system layers. However, neither of the choices of broadcast frequencies and system layers help to reach the accuracy demands of our application.

Figure 8 Effect of broadcast frequency on detecting N3 triggers (see online version for colours)

The reason for lower accuracy lies in the communication failures at the system layer. To improve the quality of communication we introduced message redundancy at the system layer. The source station broadcasts each message twice. Figure 9 shows the effect of message redundancy on the accuracy of the algorithm. There is an overall improvement in accuracy across all the system layers. The SL-250 kbps system shows improvement in low broadcast frequency areas. The reason is that a high broadcast frequency generates more traffic (messages) and because of the higher rates of packet collisions the relatively lower capacity of the

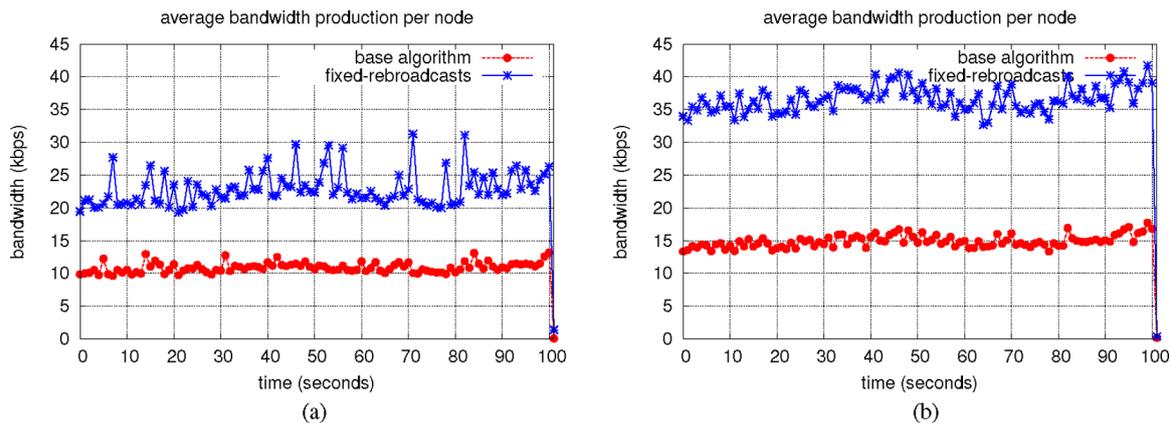
system layer masks the expected improvement due to redundancy.

Figure 9 Detecting N3 triggers with fixed-rebroadcasts (see online version for colours)

We see a decline in accuracy using SL-250 kbps after a broadcast frequency of 20. Similarly, a decline for SL-750 kbps is observed after a broadcast frequency of 30. The reason is that after these critical points, the corresponding system layers are unable to handle the generated traffic. Consequently, the accuracy starts retarding because of the higher packet collisions and packet drops at the MAC layer queues of the source stations. Figure 9 also shows that SL-750 kbps optimally uses its bandwidth to exploit message redundancy and boosts up the accuracy closer to SL-2 Mbps.

Although message redundancy helps in improving the accuracy, this improvement is not without a cost, namely, more bandwidth. Figure 10 shows a comparison of bandwidth utilisation of the base algorithm and the algorithm with message redundancy at the system layer. Note that we consider only SL-750 kbps. The improved accuracy comes at an increased cost of bandwidth consumption in case of message redundancy. Message redundancy leads to an increased rate of N3 detections, which explains the spikes in Figure 10. The bandwidth consumption is more pronounced because

Figure 10 Effect of redundancy on bandwidth consumption, using SL-750 kbps: (a) a broadcast frequency of 1 and (b) a broadcast frequency of 40 (see online version for colours)



the advertisement bundles are also broadcast twice in case of message redundancy. We also observe that the bandwidth consumption is relatively higher in the case of high broadcast frequency.

So far we assumed that the algorithm discards undecided triggers. Based on discussion in Sec. 4.4, the accuracy of the algorithm is compared with respect to three different strategies regarding the treatment of the undecided triggers. These strategies are: discard all, report all, report N2. The associated cost in terms of f^+ produced under each strategy is also assessed.

It is possible to achieve 100% accuracy by reporting all undecided triggers to CRS (see Figure 11(a)), though at the cost of a high percentage of false positives (as shown in Figure 11(b)). Discarding all undecided triggers will produce no f^+ but the corresponding level of accuracy is low. As a middle way, reporting only N2 triggers maintains higher accuracy compared to the case of discarding-all, and a lower rate of f^+ compared to the case reporting-all. Moreover, as shown in Figure 12, message redundancy raises the accuracy level to 95% for higher broadcast frequencies. The rate of f^+ is nearly constant, especially in case of reporting N2 triggers.

Now we analyse the effect of reporting N2 triggers by comparing the accuracy and efficiency across our reference system layers. Figure 13 compares the performance of the base algorithm under different system layers. Figure 13(a) shows that the accuracy level increases as we switch to system layers with higher bandwidth. The reason is that the system layer with higher bandwidth helps to detect more N2 triggers; which increases the chances of reporting those triggers that are actually N3 triggers but due to insufficient information they were promoted as only N2 triggers. On the other hand, as shown in Figure 13(b), the rate of f^+ is constant and approximately similar for all the three system layers. However, it should not be interpreted as if the same number of f^+ are reported under each system layer. For example, 20% of f^+ will mean a higher number in case more triggers are reported by a certain system layer.

Figure 14 compares the performance of the algorithm with redundancy at the system layers. The performance pattern is the same except that it is at a higher scale. The SL-2 Mbps and SL-750 kbps systems achieve an accuracy level above 95% for higher broadcast frequencies. This indicates the significance of SL-750 kbps for optimum utilisation of the bandwidth for the nature of traffic produced by the application.

7 Conclusion

Collaborative local data analysis using wireless communication is the only geographically scalable solution for high-energy cosmic ray detection. However, it suffers from false negatives due to unreliable wireless communication. Since high-energy cosmic rays are extremely rare, false negatives are unacceptable. We identified several factors that influence the number of false negatives produced by the algorithm. These include broadcast frequency, message redundancy, and handling undecided triggers. Each of these factors has an associated cost in terms of resource usage.

An increase in broadcast frequency (to a certain level) shows a positive impact on detecting the number of N3 triggers, thus decreasing the number of false negatives. This improvement comes at the cost of additional bandwidth consumption caused by message overhead due to higher number of messages. Similarly, a moderate level of message redundancy also helps reduce the number of false negatives. However, it imposes additional bandwidth requirements on the system layer. Another factor that is crucial to the performance of our algorithm is the way the undecided triggers are treated. Reporting undecided triggers to the CRS helps reduce the number of false negatives but it also causes false positives. Each false positive reported to CRS consumes resources, most importantly the communication bandwidth. We found that the best trade-off is to report N2 triggers to CRS.

In short, there are trade-offs associated with the performance of our algorithm. We identified

Figure 11 Handling undecided triggers with the base algorithm: (a) N3 triggers and (b) false positives (see online version for colours)

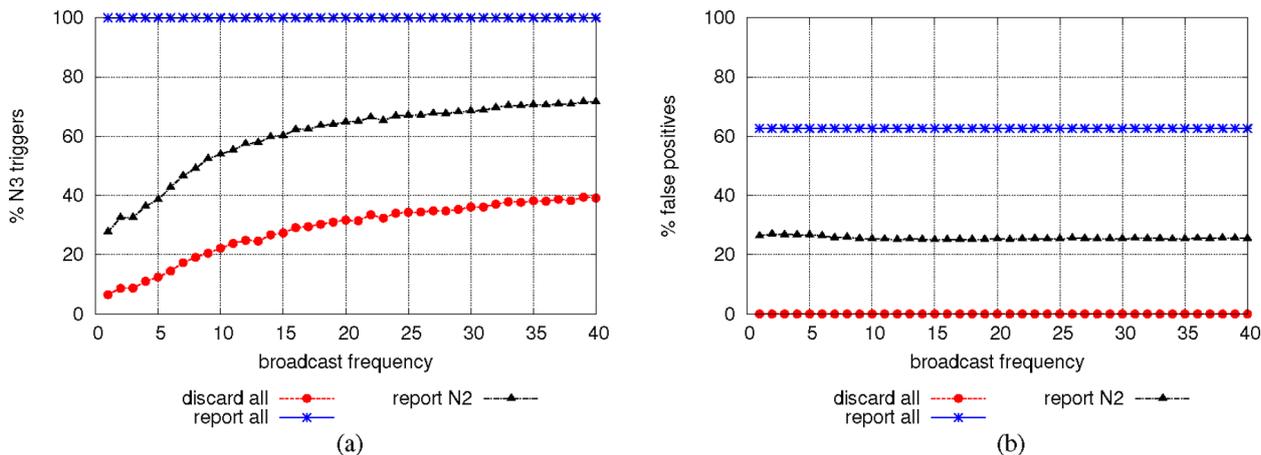


Figure 12 Handling undecided triggers with the fixed-rebroadcasts algorithm: (a) N3 detection and (b) false positives (see online version for colours)

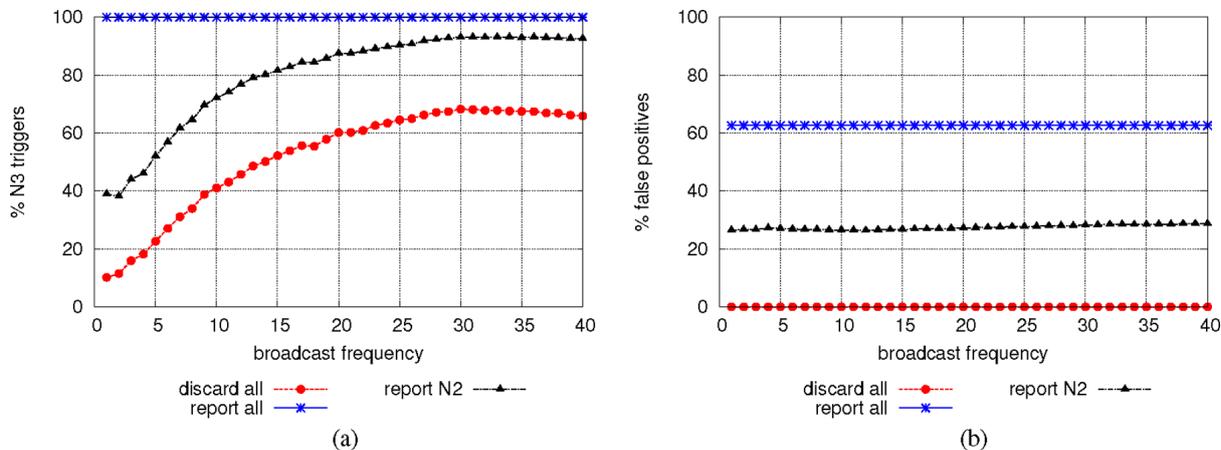
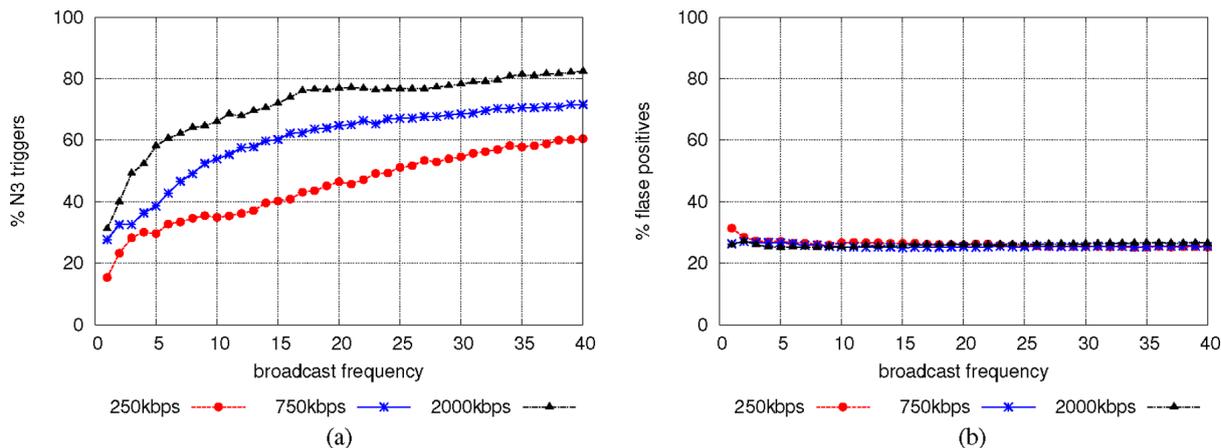


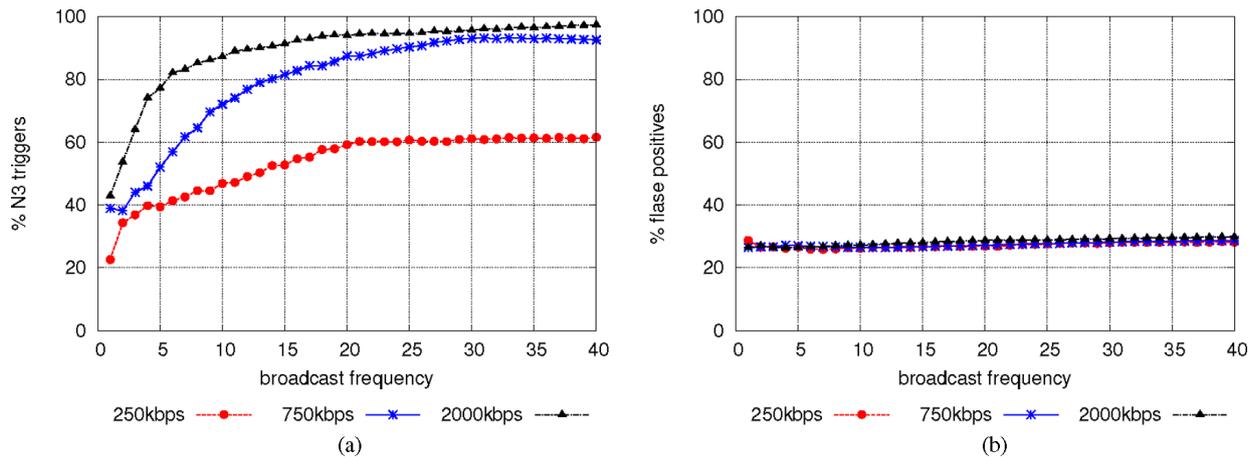
Figure 13 Reporting N2 wrt our system layers. Base algorithm, no redundancy: (a) N3 detection and (b) false positives (see online version for colours)



and experimentally analysed those trade-offs. Our experiments indicate that a carefully chosen high broadcast frequency, message redundancy at the system layer, reporting N2 triggers to CRS, and a system layer that can provide up to 750 Kbps bandwidth per station can lead to an accuracy of more than 95% at the cost of an unavoidable moderate number of false positives.

The work presented in this paper forms the core of our approach. Although we are confident that our approach is highly scalable due to its inherent nature of reliance on extremely local information. Nevertheless, we analysed the performance of our approach having a small-scale network. A challenging task for future work is to manipulate the traces without loss of its basic

Figure 14 Reporting N2 with the fixed-rebroadcasts algorithm: (a) N3 detection and (b) false positives (see online version for colours)



properties and extend them to large network on scale of thousands of nodes. Then the next step will be to analyse the performance of our algorithm in that large setup.

References

- AMS (n.d.) *Alpha Magnetic Spectrometer*, <http://ams-02project.jsc.nasa.gov/html/Projectpage.htm>
AUTHOR PLEASE SUPPLY YEAR OF PUBLICATION.
- Aramo, C. (2005) ‘Ultrahigh energy cosmic rays detection’, *AIP Conference Proceedings*, Vol. 794, No. 1, pp.240–243.
- Blümer, J. and Engel, R. & Hörandel, J.R. (2009) ‘Cosmic rays from the knee to the highest energies’, *Progress in Particle and Nuclear Physics*, Vol. 63, No. 2, pp.293–338.
- CREAM (n.d.) *The Cosmic Ray Energetics and Mass (CREAM)*, <http://cosmicray.umd.edu/cream/>
AUTHOR PLEASE SUPPLY YEAR OF PUBLICATION.
- Müller, D., Ave, M., Boyle, P.J., Gahbauer, F., Höppner, C., Hrandel, A.J., Ichimura, B.M., Müller and Romerowolf, A. (2007) ‘The TRACER project: instrument concept, balloon flights, and analysis procedures’, *Proc. Int. Cosmic Ray Conf., Vol. 2 of OG Part 1*, pp.83–86.
AUTHOR PLEASE SUPPLY LOCATION.
- Gehrke, J. and Madden, S. (2004) ‘Query processing in sensor networks’, *IEEE Pervasive Computing*, Vol. 3, pp.46–55.
- Huege, T. (2010) ‘Radio detection of cosmic rays in the Pierre Auger Observatory’, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 617, pp.484–487.
- Kelley, J. (2012) ‘Data acquisition, triggering, and filtering at the auger engineering radio array’, *Proc. VLVnT Workshop*. **AUTHOR PLEASE SUPPLY FULL DETAILS.**
- Kelley, J.L. and the Pierre Auger Collaboration (2011) ‘AERA: the auger engineering radio array’, *Proc. Int. Cosmic Ray Conf.*, arXiv:1107.4807. **AUTHOR PLEASE SUPPLY FULL DETAILS.**
- Madden, S., Franklin, M. J., Hellerstein, J.M. and Hong, W. (2002) ‘TAG: a Tiny AGgregation service for ad-hoc sensor networks’, *SIGOPS Oper. Syst. Rev.*, Vol. 36, pp.131–146.
- Martincic, F. and Schwiebert, L. (2006) ‘Distributed event detection in sensor networks’, *Proc. Conf. Systems and Networks Communication (ICSNC)*, IEEE Computer Society, pp.43–48. **AUTHOR PLEASE SUPPLY LOCATION.**
- Yousaf, S., Bakhshi, R. and van Steen, M. (2012) *Exploring Design Tradeoffs of a Distributed Algorithm for Cosmic Ray Event Detection*, Technical Report, arXiv:1209.6577v1, CoRR.
- Varga, A. (n.d.) *OMNeT++ Discrete Event Simulation System*, <http://www.omnetpp.org> **AUTHOR PLEASE SUPPLY YEAR OF PUBLICATION.**
- Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J. and Welsh, M. (2005) ‘Monitoring volcanic eruptions with a wireless sensor network’, *Proc. European Workshop on Wireless Sensor Networks (EWSN)*, IEEE, pp.108–120.
AUTHOR PLEASE SUPPLY LOCATION.
- Werner-Allen, G., Lorincz, K., Johnson, J., Lees, J. and Welsh, M. (2006) ‘Fidelity and yield in a volcano monitoring sensor network’, *Proc. Symp. on Operating systems design and implementation (OSDI)*, USENIX Association, pp.381–396. **AUTHOR PLEASE SUPPLY LOCATION.**
- Wittenburg, G., Dziengel, N., Wartenburger, C. and Schiller, J. (2010) ‘A system for distributed event detection in wireless sensor networks’, *Proc. Conf. Information Processing in Sensor Networks*, ACM, pp.94–104. **AUTHOR PLEASE SUPPLY LOCATION.**
- Woo, A., Tong, T. and Culler, D.E. (2003) ‘Taming the underlying challenges of reliable multihop routing in sensor networks’, *Proc. Conf. on Embedded Networked Sensor Systems (SenSys)*, ACM, pp.14–27. **AUTHOR PLEASE SUPPLY LOCATION.**