Strategies for Sensor Data Aggregation in Support of Emergency Response

Xianping Wang, Aaron Walden, Michele C. Weigle and Stephan Olariu
Department of Computer Science
Old Dominion University
Norfolk, VA 23529-0162
Email: {wxianpin,awalden,mweigle,olariu}@cs.odu.edu

Abstract—Sensor networks deployed in support of emergency response applications must provide timely and accurate reports of detected events. Aggregation of sensor data is required to accomplish this in an efficient manner. The aggregation problem is complicated by the fact that the perceived value of the data collected by the sensors deteriorates, often dramatically, over time. Individual sensors must determine whether to report a perceived event immediately or to defer reporting until the confidence has increased after aggregating data with neighboring nodes. However, aggregation takes time and the longer the sensors wait, the lower the value of the aggregated information. Our main contribution is a formal look at various novel aggregation strategies that relate aggregation decisions to the ensuing value of information. Our results suggest natural thresholding strategies for aggregating the information collected by groups of sensors. Our results are applicable to emergency response and other similar homeland security and military application domains where there is a strong need to model not only timely aggregation of data collected by individual actors, but also the dynamics of this aggregation. Extensive simulations have confirmed the accuracy of our theoretical predictions.

I. INTRODUCTION AND MOTIVATION

Information is a good that has value and hence can be traded [5]. Assessing the value of information has been a topic of research in economics [2], [3], [7], [16] but has only received superficial attention in sensor networks [4], [17]. It has long been recognized that, being dependent on subjective human valuation, the value of information is hard to assess. To complicate matters, it is often the case that the value of information is subject to rapid deterioration over time [1], [12].

It is widely accepted that aggregating, in a timely fashion, data collected by the various sensors that have witnessed an event is important, but measuring the value of the aggregated information remains challenging [15]. There is an obvious reason for this state of affairs: aggregation is a complex concept whose semantics are time- and context-dependent [8]. It is important to consider formal ways of defining the aggregation of time-discounted information so that aggregation might be better described and understood. In turn, this may lead to better designs for emergency response systems where, in many cases, the failure to aggregate data in a timely fashion may have catastrophic consequences [11], [12].

While it is commonly accepted that the value of information often increases as a result of aggregation, almost nothing is known about the dynamics of aggregation and about the value of the resulting information, especially in the presence of time discounting. In our previous work [14], we have investigated the interplay between time discounting and the value of aggregated information in contexts where the time discount function distributes over the aggregation operator.

The main contribution of this work is to extend the results reported in [14] to scenarios where the distributivity property does not necessarily hold. As it turns out, the fact that the distributivity property does not hold suggests aggregation strategies that, when followed, lead to maximizing the value of the aggregated information.

Our work suggests decision strategies for the aggregation of the data collected by sets of actors, whether sensors, robots, or people. Our results are applicable to fields in which hybrid networks of humans and sensors need to be formed, such as sensor networks deployed in support of emergency response and tactical applications. In both areas, there is a strong need to model not only aggregation, but also the dynamics of aggregation, as the links between different nodes shift in response to the actions of the network actors who are, in turn, shaped by the semantics of the mission at hand [13], [14].

The remainder of this work will be presented as follows. Section II provides a theoretical overview of time-discounted information, and Section III discusses aggregation of such information along with a taxonomy of possible aggregation operators. The main goal of this paper is to investigate the properties of operators (which we term Type 1) that are appropriate to use when it is safe to defer aggregation for some period of time. These Type 1 operators are discussed in Section IV. The analysis of Type 1 operators leads us to suggest techniques for thresholding, i.e., reporting an event only if the aggregated information exceeds some threshold, which are described in Section V. In Section VI, we present the results of extensive simulations that confirm the accuracy of our theoretical predictions. Finally, we conclude and present directions for future work in Section VII.

II. TIME-DISCOUNTED INFORMATION

The phenomena we discuss occur in continuous time. Consider an arbitrary sensor and let \( X \) be the random variable that denotes the “amplitude” of some sensed attribute. To
specify that the sensor has collected the data at time \( r \) we shall write \( X(r) \) and refer to it as the value of the data at time \( r \). To avoid trivialities, in this work we shall assume that \( X(r) \geq 0 \).

In general, the value of information deteriorates with time. For \( t \geq r \), the discounted value, \( X(t) \), of \( X(r) \) at time \( t \) is given by \( X(t) = X(r)\delta(t-r) \), where the map

\[ \delta : \mathbb{R}^+ \cup \{0\} \times \mathbb{R}^+ \cup \{0\} \rightarrow [0,1] \]

is referred to as a discount function. In numerous applications, \( \delta \) is a function of the difference \( t-r \) only, that is, a function of the difference between the time of data collection and the current time. In this paper we are interested in a special family of discount functions \( \delta : \mathbb{R}^+ \cup \{0\} \rightarrow [0,1] \) satisfying, for all \( r, t, (0 \leq r \leq t) \), the condition

\[ X(t) = X(r)\delta(t-r) \]  

Equation (1) tells us that the penalty of waiting for \( t-r \) time is that the value of the information collected by the sensor decreases from \( X(r) \) to \( X(t) \). Olariu et al. [14] have shown that if \( \delta \) satisfies (1) then for all \( r, s, t \) with \( 0 \leq r \leq s \leq t \),

\[ \delta(t-r) = \delta(s-r)\delta(t-s) \]  

and that for all \( r \) and \( t \) with \( 0 \leq r \leq t \), and \( \mu = -\ln \delta(1) > 0 \),

\[ \delta(t-r) = e^{-\mu(t-r)}. \]  

In other words, (1) implies an exponential deterioration of the value of data over time.

III. AGGREGATING TIME-DISCOUNTED INFORMATION

To counter-balance the effect of time discounting, we define an algebraic operation on sensor data that we call aggregation. Consider two sensors that have collected data about an event at times \( r \) and \( s \), respectively. Let \( X(r) \) and \( Y(s) \) be, respectively, the values of the information collected by the two sensors. At some later time \( \tau \), the two sensors decide to aggregate their information.

By definition, the aggregated value of \( X(r) \) and \( Y(s) \) at time \( \tau \), with \( 0 \leq r \leq t \) and \( 0 \leq s \leq t \), is \( X(t) \land Y(t) \). By (1), it follows that

\[ X(t) \land Y(t) = [X(r)\delta(t-r)] \ominus [Y(s)\delta(t-s)]. \]  

Thus, what is being aggregated at time \( t \) are the discounted values \( X(r)\delta(t-r) \) and \( Y(s)\delta(t-s) \). The aggregation operator can be extended, in the obvious way, to an arbitrary number of values that need to be aggregated.

Useful instances of \( \ominus \) include +, max, min, XOR. OR, among many others. It is worth observing that the type of aggregation operator that should be applied is application-dependent. In fact, an aggregation operator that makes sense in one application may be irrelevant in a different context.

Most of the aggregation operators \( \ominus \) of practical relevance have the following fundamental properties:

**Commutativity**: \( X(t) \ominus Y(t) = Y(t) \ominus X(t) \) for all \( t \geq 0 \). In other words, the result of the aggregation does not depend on the order in which the values are aggregated.

**Associativity**: \( [X(t) \ominus Y(t)] \ominus Z(t) = X(t) \ominus [Y(t) \ominus Z(t)] \) for all \( t \geq 0 \). If several values are aggregated in groups, the value of the aggregated information does not depend on the order in which groups are formed. We shall follow established practice to write \( X(t) \ominus Y(t) \ominus Z(t) \) instead of the cumbersome parenthesis-sized expressions. A straightforward inductive argument shows that if \( \ominus \) is associative then for an arbitrary collection of \( n \) values \( X_1(t), X_2(t), \ldots, X_n(t) \) we have

\[ [X_1(t) \ominus X_2(t) \ominus \cdots \ominus X_{n-1}(t)] \ominus X_n(t) = X_1 \ominus [X_2 \ominus [X_3 \ominus \cdots \ominus X_n(t)]] = X_1 \ominus X_2 \ominus X_3 \ominus \cdots \ominus X_n(t). \]

To simplify the notation, when aggregating several values \( X_1(t), X_2(t), \ldots, X_n(t) \), we shall write \( \bigodot_{t=1}^n X(t) \) instead of \( X_1(t) \ominus X_2(t) \ominus \cdots \ominus X_n(t) \).

**Idempotency**: If \( Y(t) = 0 \) then \( X(t) \ominus Y(t) = X(t) \). In other words, aggregation with data of value 0 has no effect.

A. A Taxonomy of Aggregation Operators

Consider, again, two sensors that have collected, at time \( t \), data about some attribute of an event they have witnessed and let \( X(t) \) and \( Y(t) \) be, respectively, the values of the information collected. It is important for \( X(t) \) and \( Y(t) \) to be aggregated in order to obtain a more reliable and, perhaps, more relevant information about the event at hand.

Suppose, further, that aggregation takes time and that the aggregated information is available at time \( \tau > t \). Given that the value of the data decays over time according to (1), and given an aggregation operator \( \ominus \), what strategy maximizes the value of the information at time \( \tau \)?

The answer to this question depends on the type of aggregation operator used. Indeed, the aggregation operator \( \ominus \) can be one of three distinct types defined as follows:

- **Type 1**: For all \( t, \tau \) with \( 0 \leq t \leq \tau \), \( [X(t) \ominus Y(t)] \ominus (\tau - t) < X(\tau) \ominus Y(\tau) \). In other words, in the case of a Type 1 operator, it is best to defer aggregation as long as the semantics of the application permit.

- **Type 2**: For all \( t, \tau \) with \( 0 \leq t \leq \tau \), \( [X(t) \ominus Y(t)] \ominus (\tau - t) = X(\tau) \ominus Y(\tau) \). As it is apparent, in the case of a Type 2 operator, it does not matter when the aggregation takes place.

- **Type 3**: For all \( t, \tau \) with \( 0 \leq t \leq \tau \), \( [X(t) \ominus Y(t)] \ominus (\tau - t) > X(\tau) \ominus Y(\tau) \). Thus, in the case of a Type 3 operator, the best strategy is to aggregate as early as the data is available and/or the semantics of the application permit.

Observe that for a Type 2 operator, the discount function distributes over \( \ominus \). Consequently, for such an operator it does
not matter whether we first aggregate and then discount the aggregated information or vice versa. Olariu et al. [14] have studied extensively Type 2 operators and their properties.

The main goal of this work is to extend the work in [14] to the case of Type 1 operators, whose properties we investigate in Section IV. Due to space limitations, Type 3 operators and the aggregation strategies they suggest, are not discussed here. They will be studied in detail in our future work.

IV. A SPECIAL TYPE 1 OPERATOR

We begin by taking note of a non-trivial Type 1 aggregation operator that turns out to have interesting applications. Imagine that the data collected by sensors take on values in the range [0, 1] and consider the aggregation operator \( \diamond \) defined as

\[
X(t) \diamond Y(t) = X(t) + Y(t) - X(t)Y(t).
\]

We leave it to the reader to verify that \( \diamond \) satisfies the associativity, commutativity and idempotency properties defined above. To prove that \( \diamond \) is a Type 1 operator, consider an arbitrary \( \tau \) with \( 0 \leq t \leq \tau \) and write

\[
[X(t) \diamond Y(t)] \delta(\tau - t)
= [X(t) + Y(t) - X(t)Y(t)] \delta(\tau - t)
= X(t) \delta(\tau - t) + Y(t) \delta(\tau - t) - X(t)Y(t) \delta(\tau - t)
= X(\tau) + Y(\tau) - X(\tau)Y(\tau) \quad [\text{by (1)}]
< X(\tau) + Y(\tau) - X(\tau)Y(\tau) \quad [\text{since } Y(\tau) < Y(t)]
= X(\tau) \diamond Y(\tau),
\]

confirming that the aggregation operator defined in (5) is of Type 1.

For later reference, we take note of the following useful property of the operator in (5), established in [14].

**Lemma 4.1:** Consider values \( X_1, X_2, \ldots, X_n \) in the range [0, 1] acted upon by the aggregator \( \diamond \) defined in (5). Then the aggregated value \( \diamond_{i=1}^n X_i \) satisfies the condition

\[
\diamond_{i=1}^n X_i = 1 - \Pi_{i=1}^n (1 - X_i).
\]

A. Reasoning about the Discounted Value of Aggregated Information

Let \( \diamond \) be an arbitrary Type 1 operator and assume that \( n, \ (n \geq 2) \) sensors have collected data about an event at times \( t_1, t_2, \ldots, t_n \). Let \( X_1(t_1), X_2(t_2), \ldots, X_n(t_n) \) be, respectively, the values collected by the sensors. At some later time \( \tau \geq \max_{1 \leq i \leq n} t_i \), the sensors decide to aggregate their information. We take note of the following relevant result.

**Lemma 4.2:** Assume an associative Type 1 operator \( \diamond \). For all \( t, \tau \) with \( \max_{1 \leq i \leq n} t_i \leq t \leq \tau \) we have

\[
[\diamond_{i=1}^n X_i(t)] \delta(\tau - t) < [\diamond_{i=1}^n X_i(t)] \delta(\tau - t).
\]

**Proof:** The proof is by induction on \( n \). For \( n = 2 \), the conclusion follows straight from the definition of a Type 1 operator. For the inductive step, let \( n \geq 3 \) be arbitrary and assume that the property holds for \( n - 1 \) aggregated values. We write

\[
[\diamond_{i=1}^n X_i(t)] \delta(\tau - t)
= [\diamond_{i=1}^{n-1} X_i(t)] \diamond X_n(t) \delta(\tau - t)
< [\diamond_{i=1}^{n-1} X_i(t)] \diamond [X_n(t) \delta(\tau - t)]
= \diamond_{i=1}^n [X_i(t) \delta(\tau - t)]
= \diamond_{i=1}^n X_i(\tau),
\]

completing the proof of Lemma 4.2.

The left-hand side of the inequality above is the discounted value of \( \diamond_{i=1}^n X_i(t) \) at time \( \tau \), while the right-hand side is the aggregated value of the discounted values of \( X_i(t) \) at time \( \tau \). Lemma 4.2 asserts that the defining inequality of the Type 1 operator holds when an arbitrary number, \( n \), of values are being aggregated.

Consider an event witnessed by \( n, \ (n \geq 2) \) sensors and let the sensed values be \( X_1, X_2, \ldots, X_n \), collected, respectively, at times \( t_1, t_2, \ldots, t_n \). Assume, further, that various groups of sensors have aggregated their data before time \( t \) and that, finally, at time \( t \) the aggregation has been completed. We are interested in evaluating the time-discounted value of the information collected by the sensors at time \( t \), where \( t \geq \max\{t_1, t_2, \ldots, t_n\} \). The answer to this natural question is provided by the following fundamental result.

**Theorem 4.3:** Assuming that the Type 1 aggregation operator \( \diamond \) is associative and commutative, the discounted value of the aggregated information at time \( t \) is upper-bounded by \( \diamond_{i=1}^n X_i(t) \), regardless of the order in which the values were aggregated.

**Proof:** The proof is by induction on \( n \). For \( n = 2 \), the conclusion follows at once from definition. Now, let \( n \geq 3 \), be arbitrary and assume the statement true for all \( m, \ (m < n) \). We assume, without loss of generality, that the last aggregation takes place at time \( t \). This aggregation must have involved a number of disjoint groups \( G_1, G_2, \ldots, G_p \) each of them the result of a previous aggregation at times, respectively, \( u_1, u_2, \ldots, u_p \). Observe that we can always re-label the groups in such a way that their aggregation times are ordered as \( u_1 < u_2 < \cdots < u_p \).

Let us look at group \( G_k \). By the induction hypothesis, the value of information in group \( G_k \) aggregated at time \( u_k \) is upper-bounded by \( \diamond_{i=1}^{u_k} X_i(u_k) \) where, of course, we assume that group \( G_k \) involves \( n_k \) sensors whose values were aggregated.

Assuming \( t \geq u_k \), Lemma 4.2 guarantees that the discounted value at time \( t \) is upper-bounded by

\[
[\diamond_{i=1}^{u_k} X_i(u_k)] \delta(t - u_k) < [\diamond_{i=1}^{u_k} X_i(t)],
\]

which is an upper bound on the value of information collected by sensors in group \( G_k \), had it been aggregated at time \( t \). Since \( G_k \) was arbitrary, the conclusion follows.

Theorem 4.3, in effect, says that the maximum value of the aggregated information that can be attained is independent of the order in which the values are aggregated. In practical terms,
Theorem 4.3 gives the algorithm designer the freedom to schedule aggregation in a random manner, much in line with the stochastic nature of wireless communication and sensor data aggregation. Recall that for Type 1 operators, aggregation can be delayed as long as the semantics of the application permit. In the next subsection we look at a thresholding mechanism that defines how long it is feasible to delay aggregation.

V. THRESHOLDING

As already mentioned, we assume that reporting a false positive involves a huge overhead and is considered prohibitively expensive. Mindful of this state of affairs, and having aggregated, at time $t$, the information collected by the various sensors, it is important to decide whether this information warrants reporting.

A. A fixed aggregation strategy

One of the natural strategies employed is thresholding. Specifically, a policy is followed of first setting up an application-dependent threshold $\Delta$ and then reporting an event only if the aggregated information exceeds $\Delta$.

Assume that $n$, ($n \geq 2$), sensors have collected data about an event at times $t_1$, $t_2$, $\ldots$, $t_n$ and let $t = \max \{t_1, t_2, \ldots, t_n\}$. Further, let $X_1(t_1), X_2(t_2), \ldots, X_n(t_n)$ be the values of the data collected by the sensors. Assuming that $\tilde{\Diamond}_{i=1}^n X_i(t) > \Delta$ the time at which the aggregation is performed is critical. We have seen that for a Type 1 operator aggregation may be delayed as long as possible. It is, however, intuitively clear that if aggregation is delayed too much, the aggregated value might not exceed the threshold and a relevant event would go unreported. Thus, the question is to determine the time window during which the sensors need to aggregate their values in order for the aggregated value to exceed the threshold $\Delta$.

Let $\tau$ be the latest time at which aggregation should be performed. Since we are interested in Type 1 operators, it is natural to insist that $\tilde{\Diamond}_{i=1}^n X_i(t) \geq \Delta$. Recalling that by (3), $\delta(t - \tau) = e^{-\mu(t-\tau)}$, the above inequality becomes $e^{-\mu(t-\tau)} > \frac{\Delta}{\tilde{\Diamond}_{i=1}^n X_i(t)}$ which, upon taking logarithms on both sides, yields

$$-\mu(t - \tau) \geq \ln \frac{\Delta}{\tilde{\Diamond}_{i=1}^n X_i(t)}.$$

Upon solving for $\tau$ we obtain

$$\tau < t + \frac{1}{\mu} \ln \frac{\Delta}{\tilde{\Diamond}_{i=1}^n X_i(t)}.$$

As the last data was collected at time $t = \max \{t_1, t_2, \ldots, t_n\}$, (6) specifies that, past $t$, there is a time window of size $\frac{1}{\mu} \ln \frac{\Delta}{\tilde{\Diamond}_{i=1}^n X_i(t)}$ during which aggregation must occur. This result gives the system designer a handle on the types of aggregation protocols to use.

B. An adaptive aggregation strategy

As before, sensor readings about an event were collected and the resulting values $X_1, X_2, \ldots$ are reals in $[0,1]$. Assume that one of the network actors (e.g., a sensor) is in charge of the aggregation process and that the operator $\Diamond$ defined in (5) is employed in conjunction with a threshold $\Delta > 0$.

We now state and prove a technical result that will motivate our adaptive aggregation strategy.

**Theorem 5.1:** If $X_{i_1}, X_{i_2}, \ldots, X_{i_m}, m > 1$, satisfy $X_{i_j} > 1 - \sqrt[3]{1-\Delta}, j = 1, 2, \ldots, m$, then $\tilde{\Diamond}_{j=1}^m X_{i_j} > \Delta$.

**Proof:** By Lemma 4.1,

$$\tilde{\Diamond}_{j=1}^m X_{i_j} = 1 - \Pi_{j=1}^m (1 - X_{i_j}) > 1 - \Pi_{j=1}^m \left(1 - (1 - \sqrt[3]{1-\Delta}) \right) = 1 - \Pi_{j=1}^m \sqrt[3]{1-\Delta} = 1 - (1 - \Delta) = \Delta.$$

Notice what Theorem 5.1 says: if there are two sensors whose individual values exceed $1 - \sqrt[3]{1-\Delta}$, then the two should aggregate their values and, having exceeded $\Delta$, should report the event. Similarly, if there are three sensors whose individual values exceed $1 - \sqrt[3]{1-\Delta}$, then the result of their aggregated data exceeds $\Delta$, etc.

In turn, this observation suggests the following adaptive aggregation strategy: in the first aggregation round, the aggregator will announce the target $1 - \sqrt[3]{1-\Delta}$. If at least two sensors (including the aggregator) hold values in excess of the target $1 - \sqrt[3]{1-\Delta}$, then the event will be reported. If the first round of aggregation suffices, all is well. If, however, there is an insufficient number of sensors holding suitable values, the second round begins. In this round, the aggregator announces the target $1 - \sqrt[3]{1-\Delta}$. If three or more sensors can be identified that exceed this target, then by Theorem 5.1 the aggregated value must exceed $\Delta$ and so the event is reported. This aggregation strategy is continued, as described, until either an event is reported, or else the results are inconclusive and no event is reported.

![Fig. 1. Illustrating a fire event.](image-url)
times $t_1 < t_2 < t_3 < t_4$. Further, let $X_1(t_1), X_2(t_2), X_3(t_3), X_4(t_4)$ and $X_5(t_5)$ be the values thus collected, normalized to $[0,1]$. Given the layout of the sensors, it is reasonable to assume that $X_1(t_1) \geq X_2(t_2) \geq X_3(t_3) \geq X_4(t_4) \geq X_5(t_5)$. Since sensor $B$ is closest to the fire it will be the first one to sense high temperature and, thus, will become an aggregator. $B$ will wait for other sensors to report a reading over $1 - \sqrt{1-\Delta}$ and will attempt aggregation with such sensors.

![Graph](image)

**Fig. 2. Illustrating our adaptive aggregation strategy.**

Referring to Figure 2, the first two rounds of our adaptive aggregation strategy do not yield a sufficient number of sensors to effect an aggregation. In the third round, $B$ announces the target $1 - \frac{\sqrt{1-\Delta}}{\mu}$. This third round yields four sensors whose individual values exceed the announced target and, consequently, the fire event will be reported.

Notice that with each round, the value of the information decays. When the aggregator and other sensors wait for more sensors to confirm, their value will decay. In order that their values are all above the final threshold, their time interval should satisfy the following constraints.

$$t_k - t_i < \frac{1}{\mu} \ln \frac{1 - \sqrt{1-\Delta}}{1 - \frac{\sqrt{1-\Delta}}{\mu}}, \quad 1 \leq i < k \leq n,$$

while $X_1(t_1) > X_2(t_2) > \ldots > X_n(t_n)$.

In practice, the number $n$ of sensors for aggregation is a function of the deployment density and, in general, is application specific.

**VI. SIMULATION RESULTS**

Imagine that fire just broke out on a ship instrumented by a set of relevant sensors, including temperature and humidity sensors, light and smoke detectors, etc. [6]. To keep the aggregation operator simple, in what follows we assume the existence of temperature sensors only. Aggregating data across sets of sensors detecting different attributes of a fire event would proceed along similar lines, but is not covered here.

Imagine a critical temperature range $K = [100,1000]^\circ C$. The various sensors take temperature readings $T_1, T_2, \ldots$. The value $X_i$ associated with temperature $T_i$ is defined to be $X_i = \Pr[T_i \in K | F]$. That is, the conditional probability of temperature $T_i$ being recorded by a sensor, given the event $F$ that fire is present. In order to stamp out noise, $X_i$ is taken to be 0.9 if $T_i$ is in the critical range $K$ and 0 otherwise.

We define the aggregator $\diamond$ as follows

$$X_i \diamond X_j = \Pr\{T_i \in K \cup \{T_j \in K\} | F\}. \quad (7)$$

In other words, aggregating two values is tantamount to computing the conditional probability of a union of events. Assuming that the sensors act independently, (7) implies that

$$X_i \diamond X_j = P\{T_i \in K \cup \{T_j \in K\} | F\} = \Pr\{T_i \in K | F\} + \Pr\{T_j \in K | F\} - \Pr\{T_i \in K \cap \{T_j \in K\} | F\} = \Pr\{T_i \in K | F\} + P\{T_j \in K | F\} - \Pr\{T_i \in K \cap \{T_j \in K\} | F\} = X_i + X_j - X_i X_j,$$

confirming that the aggregation operator $\diamond$ is the one defined in (5). One can also see that $\diamond$ has the property that the value of the aggregated information increases with the number of sensor readings in the critical temperature range.

In the following simulations using MATLAB, the distribution of temperature in the fire is approximated by a linear model [9] with a plateau temperature of $1000^\circ C$ and an ambient temperature of $20^\circ C$. The critical temperature range is $[100,1000]^\circ C$, and the value discount constant $\mu$ is $0.15^{-1}$ [14]. The fire propagation model is approximated by a dot source spreading out at the same rate in all directions at a speed of 1m/s [9]. The fire front has a temperature of $1000^\circ C$ that decreases to the ambient temperature within 3m. The temperature sensors are deployed in rectangular lattices of size $3 \times 2$ in a plane with every side of 3m (Figure 3). The fire source location is randomly generated in one of the rectangles. Due to the small distances involved, wireless communication delays are ignored. We assume a sensor sampling time of 2s, and that the sensors are asynchronous, i.e., a new temperature reading will be reported every 2 seconds.

When a fire occurs and disseminates, every sensor senses a rising temperature and computes the corresponding value. Figure 4 illustrates this process for a single sensor. All other sensors behave in a similar manner. The dots are sensed temperatures, sampled every 2s. The solid line is the corresponding value, which jumps from 0 to 0.9 when the sensed temperature

---

1. This ordering is assumed for illustration purposes but is not really necessary.
enters the critical range. It then decays until the next sample and the value is renewed. The process repeats to form the shape of a saw-tooth.

With fixed thresholding, the fire alarm is set if the aggregated value of three sensors is above a threshold of 0.99. In Figure 5, the top six lines that oscillate between 0.7 to 0.9 are the values of six sensors’ temperature. The middle panel shows the aggregated value with fixed thresholding. The alarm is triggered repeatedly every time the aggregated value rises above the threshold $\Delta$.

With fixed thresholding, a base station is required. We can achieve the same triggering of the alarm without a base station by using adaptive thresholding. Here, the initial threshold $\Delta_0$ is 0.8, and the final threshold $\Delta$ is 0.99. In this case, the maximum number of sensors needed for reporting a fire event is five. The bottom panel in Figure 5 shows the adaptive aggregated value. Once a single sensor’s value rises above the initial threshold, it becomes the aggregator (in this case, sensor $B$). As new readings are shared, the aggregated value rises above the 0.99 threshold and the alarm is triggered. As can be seen, the pattern of alarms in the bottom panel (adaptive thresholding) is similar to that in the middle panel (fixed thresholding), without the need for a base station performing the aggregation.

VII. CONCLUSIONS AND FUTURE WORK

This paper provided a formal way of looking at aggregation of information in networks where individual actors possess information whose value decays over time. We offered a formal model for the valuation of time-discounted information and of the algebra of its aggregation.

We allowed aggregation of time-discounted information to proceed in an arbitrary, not necessarily pairwise, manner. We have shown that the resulting value of the aggregate does not depend on the order in which aggregation of individual values take place. Our results suggest natural thresholding strategies for the aggregation of the information collected by sets of network actors. Our theoretical predictions were confirmed by extensive simulation.

In spite of these results, a number of problems are open and are getting attention. First, if an emergency event is witnessed by sensors detecting different attributes of the event, it is of interest to aggregate data across various types of sensors in a cooperative fashion.

Second, it is of great theoretical and practical interest to consider other time discounting regimens that do not conform to (1). Examples of such regimens are known to exist, step functions being a prime example. The question that we are addressing here is that of approximating a step function and, indeed, other similar decay functions by polynomials. It is a classic result of Weierstrass that every real function can be approximated by a suitable sequence of polynomials [10]. This approach seems natural as every exponential can be approximated by a polynomial consisting of the first few terms in its Taylor expansion.

Finally, it is of practical relevance to retask the sensors as the mission dynamics evolve. Retasking may involve moving from one set of sensed attributes to another and inferring an attribute for which the sensor does not have a direct sensing capability.

Addressing these, and other similar related open problems, promises to be an exciting area of work.

REFERENCES


