Delay Optimal Event Detection on Ad Hoc Wireless Sensor Networks

PREMKUMAR KARUMBU, VENKATA K. PRASANTHI,
and ANURAG KUMAR, Indian Institute of Science, Bangalore

12

We consider a small extent sensor network for event detection, in which nodes periodically take samples and then contend over a *random access network* to transmit their measurement packets to the fusion center. We consider two procedures at the fusion center for processing the measurements. The Bayesian setting, is assumed, that is, the fusion center has a prior distribution on the change time. In the first procedure, the decision algorithm at the fusion center is *network–oblivious* and makes a decision only when a complete vector of measurements taken at a sampling instant is available. In the second procedure, the decision algorithm at the fusion center is *network–aware* and processes measurements as they arrive, but in a time-causal order. In this case, the decision statistic depends on the network delays, whereas in the network–oblivious case, the decision statistic does not. This yields a Bayesian change-detection problem with a trade-off between the random network delay and the decision delay that is, a higher sampling rate reduces the decision delay but increases the random access delay. Under periodic sampling, in the network–oblivious case, the structure of the optimal stopping rule is the same as that without the network, and the optimal change detection delay decouples into the network delay and the optimal decision delay without the network. In the network–aware case, the optimal stopping problem is analyzed as a partially observable Markov decision process, in which the states of the queues and delays in the network need to be maintained. A sufficient decision statistic is the network state and the posterior probability of change having occurred, given the measurements received and the state of the network. The optimal regimes are studied using simulation.

Categories and Subject Descriptors: C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network monitoring*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Control theory*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Optimal change detection over a network, detection delay, cross–layer design of change detection

ACM Reference Format:
Karumbu, P., Prasanthi, V. K., and Kumar, A. 2012. Delay optimal event detection on ad hoc wireless sensor networks. ACM Trans. Sensor Netw. 8, 2, Article 12 (March 2012), 35 pages.
DOI = 10.1145/2140522.2140525 http://doi.acm.org/10.1145/2140522.2140525

1. INTRODUCTION

A wireless sensor network is formed by tiny, untethered devices (motes) that can sense, compute, and communicate. Sensor networks have a wide range of applications such as environment monitoring, detecting events, identifying locations of survivors in building

---

This is an expanded version of a paper that was presented at *SECON'06*.
This work was supported in part by grant number 2900 IT from the Indo-French Center for the Promotion of Advanced Research (IFCPAR) and in part by a project from DRDO, Government of India.
Authors' addresses: P. Karumbu, V. Prasanthi, and A. Kumar, Dept. of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India; email: {kprem, anurag}@ece.iisc.ernet.in and prasanthi.m@gmail.com.

DOI 10.1145/2140522.2140525 http://doi.acm.org/10.1145/2140522.2140525

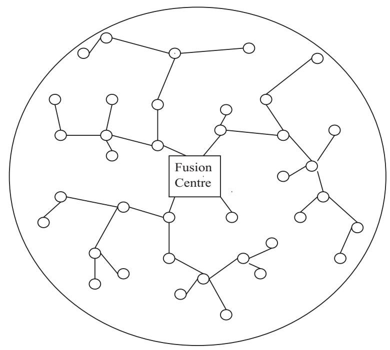

Fig. 1. An ad hoc wireless sensor network with a fusion center is shown. The small circles are the sensor nodes (motes), and the lines between them indicate wireless links obtained after a self-organization procedure.

and train disasters, and intrusion detection for defense and security applications. For factory- and building-automation applications, there is increasing interest in replacing wireline sensor networks with wireless sensor networks, due to the potential reduction in costs of engineering, installation, operations, and maintenance [Honeywell Inc.; ISA].

Event detection is an important task in many sensor network applications. In general, an event is associated with a change in the distribution of a related quantity that can be sensed. For example, the event of a gas leakage at any joint in a pipe causes a change in the distribution of pressure at the joint and, hence, can be detected with the help of pressure sensors. In this article, we limit our discussion to the centralized fusion model (see Figure 1), in which each mote in an event detection network senses and sends some function of its observations (e.g., quantized samples) to the fusion center at a particular rate. The fusion center, by appropriately processing the sequence of values it receives, makes a decision regarding the state of nature, that is, it decides whether or not a change has occurred.

Our problem is that of minimizing the mean detection delay (the delay between the event occurring and the detection decision at the fusion center) with a bound on the probability of false alarm. We consider *a small extent network* in which all the sensors have the same coverage, that is, when the change in distribution occurs, it is observed by all the sensors, and the statistics of the observations are the same at all the sensors. N sensors synchronously sample their environment at a particular rate. Synchronized operation across sensors is practically possible in networks such as 802.11 WLANs and Zigbee networks, since the access point and the PAN coordinator, respectively, transmit beacons that provide all nodes with a time reference. Based on the measurement samples, the nodes send certain values (e.g., quantized samples) to the fusion center. Each value is carried by a packet, which is transmitted using a contention-based multiple access mechanism. Thus, our small extent network problem is a natural extension of the standard change-detection problem (see Veeravalli [2001] and the references therein) to detection over a random access network. The problem of quickest event detection in a large extent network (where the region of interest is much larger than the sensing coverage of any sensor) is considered by us in Premkumar et al. [2009]. Also, a small extent network can be thought of as a cluster in a large extent network, and the decision maker can be thought of as a cluster head.

In this setting, due to the multiple access network delays between the sensor nodes and the fusion center, several possibilities arise. In Figure 2, we show that although the sensors take samples synchronously, due to random access delays, the various packets sent by the sensors arrive at the fusion center asynchronously. As shown in the figure,

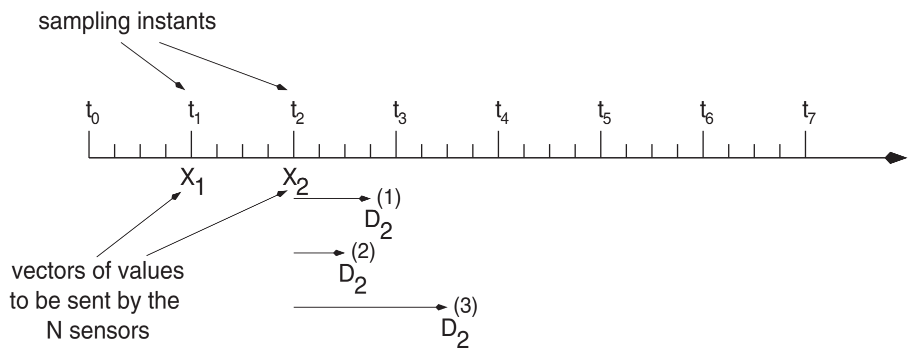

Fig. 2. The sensors periodically take samples at instants $t_1, t_2, \ldots$, and prepare to send a vector of values $\mathbf{X}_h = [X_h^{(1)}, X_h^{(2)}, \ldots, X_h^{(N)}]$ at t_h to the fusion center. Each sample is queued as a packet in the queue of the respective node. Due to multiple access delays, the packets arrive with random delays at the fusion center, for example, for $\mathbf{X}_2$, the delays $D_2^{(1)}, D_2^{(2)}$, and $D_2^{(3)}$ for the packets from sensors $1, 2$, and 3 are shown.

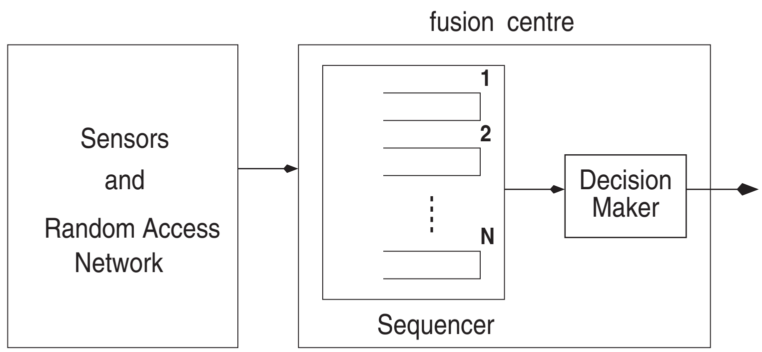

Fig. 3. A conceptual block diagram of the wireless sensor network shown in Figure 1. The fusion center has a sequencing buffer which queues out-of-sequence samples and delivers the samples to the decision maker in time-order, as early as possible, batchwise or samplewise.

the packets generated due to the samples taken at time t_2 arrive at the fusion center with a delay of $D_2^{(1)}, D_2^{(2)}, D_2^{(3)}$, etc. It is even possible that a packet corresponding to the samples taken at time t_3 can arrive before one of the packets generated due to the samples taken at time t_2.

Figure 3 depicts a general queueing and decision-making architecture in the fusion center. All samples are queued in per-node queues in a sequencer. The way the sequencer releases the packets gives rise to the following three cases, the first two of which we study in this article.

(1) The sequencer queues the samples until all the samples of a batch (a *batch* is the set of samples generated at a sampling instant) are accumulated; it then releases the entire batch to the decision device. The batches arrive to the decision maker in a time-sequence order. The decision maker processes the batches without knowledge of the state of the network (i.e., reception times at the fusion center and the numbers of packets in the various queues). We call this *Network Oblivious Decision Making* (NODM). In factory and building automation scenarios, there is a major impetus to replacing wireline networks between sensor nodes and controllers. In such applications, the first step could be to retain the fusion algorithm in the controller, while replacing the wireline network with a wireless ad hoc network.

Indeed, we show that this approach is optimal for NODM, provided the sampling rate is appropriately optimized.

(2) The sequencer releases samples only in time-sequence order but does not wait for an entire batch to accumulate. The decision maker processes samples as they arrive. We call this *Network Aware Decision Making* (NADM). In NADM, whenever the decision maker receives a sample, it has to roll back its decision statistic to the sampling instant, update the decision statistic with the received sample, and then update the decision statistic to the current time slot. The decision maker makes a Bayesian update on the decision statistic, even if it does not receive a sample in a slot. Thus, NADM requires a modification in the decision-making algorithm in the fusion center.

(3) The sequencer does not queue any samples. The fusion center acts on the values from the various sampling instants as they arrive, possibly out of order. The formulation of such a problem would be an interesting topic for future research.

Our Contributions. We find that the existing literature on sequential change-detection problems (see the following discussion on related literature) assumes that, at a sampling instant, the samples from all the sensors reach the fusion center instantaneously. As already explained, however, the delay in detection in our problem is not only due to the detection procedure requiring a certain amount of samples in order to make a decision (which we call *decision delay*), but also due to the random packet delay in the multiple access network (which we call *network delay*). We work with a formulation that accounts for both these delays, while limiting ourselves to the particular fusion center behaviors explained in cases (1) and (2).

In Section 2, we discuss the basic change-detection problem and set up the model. In Section 3, we formulate the change-detection problem over a random access network in a way that naturally includes the network delay. We show that in the case of NODM, the problem objective decouples into a part involving the network delay and a part involving the optimal decision delay under the condition that the sampling instants are periodic. Then, in Section 4, we consider the special case of a network with a star topology, that is, all nodes are one hop away from the fusion center, and provide a model for contention in the random access network. In Section 5, we formulate the NADM problem, where we process the samples as they arrive at the fusion center but in a time causal manner. The out-of-sequence packets are queued in a sequencing buffer and are released to the decision maker as early as possible. We show in the NADM case that the change-detection problem can be modeled as a Partially Observable Markov Decision Process (POMDP). We show that sufficient statistics for the observations include the *network-state* (which includes the queue lengths of the sequencing buffer, network-delays) and the posterior probability of change having occurred given the measurements received and the network states. As usual, the optimal policy can be characterized via a Bellman equation, which can then be used to derive insights into the structure of the policy. We show that the optimal policy is a threshold on the posterior probability of change, and that the threshold, in general, depends on the network state. Finally, in Section 6 we compare numerically the mean detection-delay performance of NODM and a simple heuristic algorithm motivated by NADM processing. We show the trade-off between the sampling rate r and the mean detection delay. Also, we show the trade-off between the number of sensors and the mean detection delay.

Related Literature. The basic mathematical formulation in this article is an extension of the classical problem of sequential change detection in a Bayesian framework. The centralized version of this problem was solved by Shiryaev [1978]. The decentralized version of the problem was introduced by Tenny and Sandell [1981]. In the decentralized setting, Veeravalli [2001] provided optimal decision rules for the sensors and the

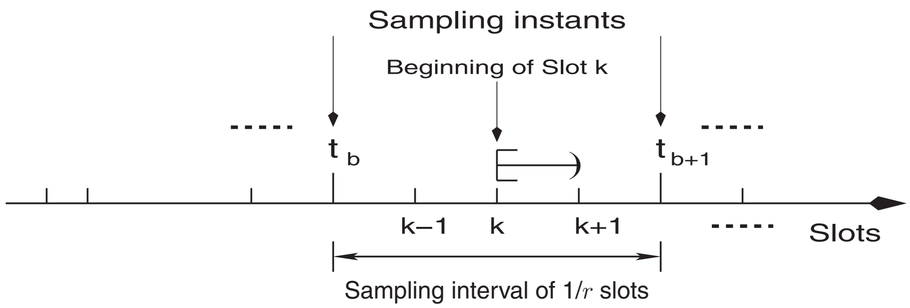

Fig. 4. Time evolves over slots. The length of a slot is assumed to be unity. Thus, slot k represents the interval $[k, k+1)$, and the beginning of slot k represents the time instant k. Samples are taken periodically every $1/r$ slots, starting from $t_1 = 1/r$.

fusion center, in the context of conditionally independent sensor observations and a quasi-classical information structure. For a large network setting, Niu and Varshney [2005] studied a simple hypothesis testing problem and proposed a counting rule based on the number of alarms. They showed that, for a sufficiently large number of sensors, the detection performance of the counting rule is close to that of the optimal rule. In a recent article on anomaly detection in wireless sensor networks Rajasegarar et al. [2008], have provided a survey of statistical- and machine-learning-based techniques for detecting various types of anomalies such as sensor faults, security attacks, and intrusions. In Aldosari and Moura [2004], the authors consider the problem of decentralized binary hypothesis testing, where the sensors quantize the observations and the fusion center makes a binary decision between the two hypotheses.

Remark. In the existing literature on the topic of optimal sequential event detection in wireless sensor networks, to the best of our knowledge, there has been no prior formulation that incorporates multiple access delay between the sensing nodes and the fusion center. Interestingly, in this article we introduce what can be called a *cross-layer* formulation involving *sequential decision theory* and *random access network delays.* In particular, we encounter the *fork-join queueing model* (see, e.g., Baccelli and Makowski [1990]) that arises in distributed computing literature.

2. THE BASIC CHANGE-DETECTION PROBLEM

In this section, we introduce the model for the basic change-detection problem. The notation we follow is given here.

- —Time is slotted, and the slots are indexed by $k = 0, 1, 2, \ldots$. We assume that the length of a slot is unity and that slot k refers to the interval $[k, k+1)$. Thus, the beginning of slot k indicates the time instant k (see Figure 4).
- —N sensors are synchronously sampling at the rate r samples/slot, that is, the sensors make an observation every $1/r$ slots and send their observations to the fusion center. For example, if $r = 0.1$, then a sample is taken by a sensor every 10^{th} slot. We assume that $1/r$ is an integer. The sampling instants are denoted $t_1, t_2, \ldots$ (see Figure 5). Define $t_0 = 0$; note that the first sample is taken at $t_1 = 1/r$.
- —The vector of network delays of batch b is denoted by

$$\mathbf{D}_b = \left[D_b^{(1)}, D_b^{(2)}, \ldots, D_b^{(N)}\right],$$

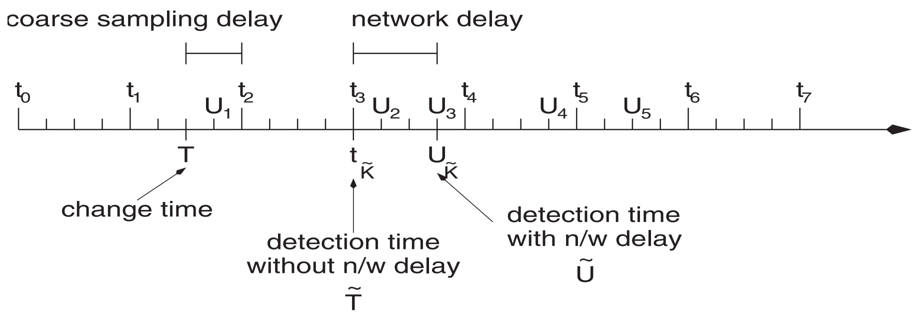

Fig. 5. Change time and detection instants with and without network delay are shown. The coarse sampling delay is given by $t_K - T$, where t_K is the first sampling instant after change, and the network delay is given by $U_{\tilde{K}} - t_{\tilde{K}}$.

where $D_b^{(i)} \in \{1, 2, 3, \ldots\}$ is the network delay in slots of the ith component of the bth batch (sampled at $t_b = b/r$). Also, note that $D_b^{(i)} \geqslant 1$, as it requires one time slot for the transmission of a packet to the fusion center after a successful contention.

—The state of nature $\Theta \in \{0, 1\}$. 0 represents the state "before change" and 1 represents the state "after change". It is assumed that the change time T (measured in slots) is geometrically distributed, that is,

$$\begin{aligned} \mathsf{P}(T = 0) &= \rho, \\ \text{and, for } k \geqslant 1, \quad \mathsf{P}(T = k \mid T > 0) &= p(1-p)^{(k-1)}. \end{aligned} \tag{1}$$

The value of 0 for T accounts for the possibility that the change took place before the observations were made.

—The vector of outputs from the sensor devices at the bth batch is denoted by

$$\mathbf{X}_b = \left[X_b^{(1)}, X_b^{(2)}, \ldots, X_b^{(N)}\right],$$

where $X_b^{(i)} \in \mathcal{X}$ is the bth output at the ith sensor. Given the state of nature, $X_b^{(i)}$s are assumed to be (conditionally) independent across sensors and independent and identically distributed over sampling instants with probability distributions $F_0(x)$ and $F_1(x)$ before and after the change, respectively. $\mathbf{X}_1$ corresponds to the first sample taken. In this work, we do not consider the problem of optimal processing of sensor measurements to yield the sensor outputs, for example, optimal quantizers (see Veeravalli [2001]).

—Let S_b, $b \geqslant 1$, be the state of nature at the bth sampling instant and S_0 be the state at time 0. Then $S_b \in \{0, 1\}$ with

$$\mathsf{P}(S_0 = 1) = \rho = 1 - \mathsf{P}(S_0 = 0).$$

S_b evolves as follows. If $S_b = 0$ for $b \geqslant 0$, then

$$S_{b+1} = \begin{cases} 1 & \text{w.p.} \quad p_r \\ 0 & \text{w.p.} \quad (1-p_r) \end{cases},$$

where $p_r = 1 - (1-p)^{1/r}$. Further, if $S_b = 1$, then $S_{b+1} = 1$. Thus, if $S_0 = 0$, then there is a change from 0 to 1 at the Kth sampling instant, where K is geometrically distributed. For $b \geqslant 1$,

$$\mathsf{P}(K = b) = p_r(1-p_r)^{b-1}.$$

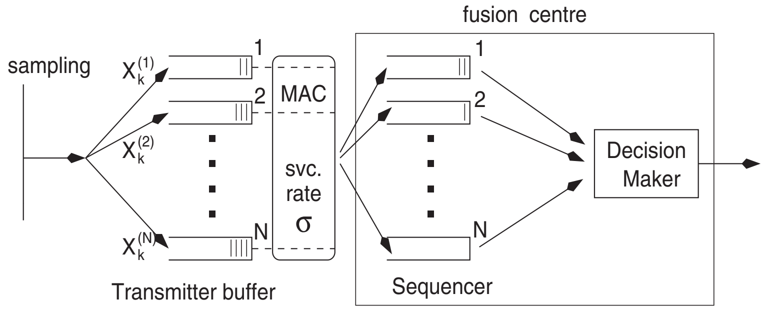

Fig. 6. A sensor network model of Figure 3 with one-hop communication between the sensor nodes and the fusion center. The random access network, along with the sequencer, is a fork–join queueing model.

Each value to be sent to the fusion center by a node is inserted into a packet, which is queued for transmission. It is then transmitted to the fusion center by using a contention-based multiple-access protocol. A node can directly transmit its observation to the fusion center or route it through other nodes in the system. Each packet takes a time slot to transmit. The MAC protocol and the queues evolve over the same time slots. The fusion center makes a decision about the change, depending on whether Network Oblivious (NODM) processing or Network Aware (NADM) processing is employed at the fusion center. In the case of NODM processing, the decision sequence (also called the *action sequence*) is $A_u,\ u \geqslant 0$, with $A_u \in$ {stop and declare change(1), take another sample(0)}, where u is a time instant at which a complete batch of N samples corresponding to a sampling instant is received by the fusion center. In the case of NADM processing, the decision sequence is $A_k,\ k \geqslant 0$, with $A_k \in$ {stop and declare change(1), take another sample(0)}, that is, a decision about the change is taken at the beginning of every slot.

3. NETWORK OBLIVIOUS DECISION MAKING (NODM)

From Figure 2, we note that although all the components of a batch b are generated at $t_b = b/r$, they reach the fusion center at times $t_b + D_b^{(i)}, i = 1, 2, \ldots, N$. In NODM processing, the samples, which are successfully transmitted, are queued in a sequencing buffer as they arrive (see Figure 6), and the sequencer releases a (complete) batch to the decision maker as soon as all the components of a batch arrive. The decision maker makes a decision about the change at the time instants when a (complete) batch arrives at the fusion center. In Network Oblivious (NODM) processing, the decision maker is oblivious to the network and processes the batch as though it had just been generated (i.e., as if there were no network, hence the name Network Oblivious Decision Making). We further define (see Figure 5) the following.

—$U_b, (b \geqslant 1)$. The random instant at which the fusion center receives the complete batch $\mathbf{X}_b$.
—$\widetilde{K} \in \{0, 1, \ldots\}$. The batch index at which the decision takes place if there is no network delay. $\widetilde{K} = 0$ means that decision 1 (stop and declare change) is taken before any batch is generated.
—$\widetilde{T} = t_{\widetilde{K}}$. The random time (a sampling instant) at which the fusion center stops and declares change if there was no network delay.
—$\widetilde{U} = U_{\widetilde{K}}$. The random time slot at which the fusion center stops and declares change in the presence of network delay.
—$D_b = U_b - t_b$. Sojourn time of the bth batch, that is, the time taken for all the samples of the bth batch to reach the fusion center. Note that D_b is given by

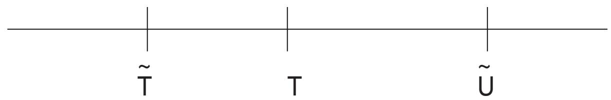

Fig. 7. Illustration of an event of false alarm with $\widetilde{T} < T$, but $\widetilde{U} > T$.

$\max\{D_b^{(i)} : i = 1, 2, \ldots, N\}$. Thus, the delay of batch $\widetilde{K}$, at which the detector declares a change, is $U_{\widetilde{K}} - t_{\widetilde{K}} = \widetilde{U} - \widetilde{T}$.

We define the following detection metrics.

Mean Detection Delay is defined as the expected number of slots between the change point T and the stopping time instant $\widetilde{U}$ in the presence of coarse sampling and network delays, that is, *Mean Detection Delay* $= \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right]$.

Mean Decision Delay is defined as the expected number of slots between the change point T and the stopping time instant $\widetilde{T}$ in the (presence of coarse sampling delay and in the) absence of network delay, that is, *Mean Decision Delay* $= \mathsf{E}\left[(\widetilde{T} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right]$.

With the preceding model and assumptions, we pose the following NODM problem: Minimize the mean detection delay with a bound on the probability of false alarm, such that the decision epochs are the time instants when a complete batch of N components corresponding to a sampling instant is received by the fusion center. In Section 5, we pose the problem of making a decision at every slot based on the samples as they arrive at the fusion center. Motivated by the approach in Veeravalli [2001], we use the following formulation for a given sampling rate r.

$$\begin{aligned} &\min \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right], \\ \text{such that} \quad &\mathsf{P}(\widetilde{T} < T) \leqslant \alpha, \end{aligned} \tag{2}$$

where α is the constraint on the false alarm probability.

Remark 3.1. Note that if $\alpha \geqslant 1-\rho$, then the decision making procedure can be stopped, and an alarm can be raised even before the first observation. Thus, we assume that $\alpha < 1 - \rho$.

Remark 3.2. Note that here we consider $\mathsf{P}(\widetilde{T} < T)$ as the probability of false alarm and not $\mathsf{P}(\widetilde{U} < T)$, that is, a case as shown in Figure 7 is considered a false alarm. This can be understood as follows. When the decision unit detects a change at slot $\widetilde{U}$, the measurements that triggered this inference would be carrying the time stamp $\widetilde{T}$, and we infer that the change actually occurred at or before $\widetilde{T}$. Thus, if $\widetilde{T} < T$, it is an error.

We write the problem defined in Equation (2) as

$$\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right], \tag{3}$$

where Π_α is the set of detection policies for which $\mathsf{P}(\widetilde{T} < T) \leqslant \alpha$.

THEOREM 1. *If the sampling is periodic at rate r and the batch sojourn time process D_b, $b \geqslant 1$ is stationary with mean $d(r)$, then*

$$\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right] = (d(r) + l(r))(1 - \alpha) - \rho \cdot l(r) + \frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+,$$

where $l(r)$ is the delay due to (coarse) sampling.

Remark 3.3. For example, in Figure 5, the delay due to coarse sampling is $t_2 - T$, $\widetilde{K} - K = 3 - 2 = 1$, and the network delay is $U_3 - t_3$. The stationarity assumption on

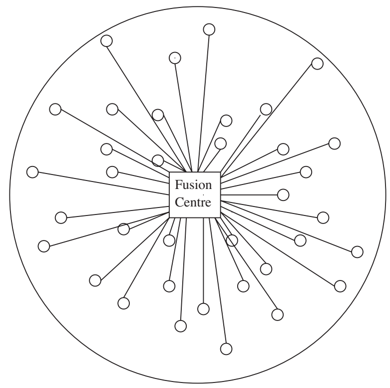

Fig. 8. A sensor network with a star topology with the fusion center at the hub. The sensor nodes use a random access MAC to send their packets to the fusion center.

D_b, $b \geqslant 1$ is justifiable in a network in which measurements are continuously made, but the detection process is started only at certain times, as needed.

Proof. The following is a sketch of the proof (the details are in Appendix I).

$$\begin{aligned}\min_{\Pi_\alpha} \mathsf{E}\big[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\big] &= \min_{\Pi_\alpha} \left\{\mathsf{E}\big[(\widetilde{U} - \widetilde{T})\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\big] + \mathsf{E}\big[\widetilde{T} - T\big]^+\right\} \\ &= \min_{\Pi_\alpha}\{\mathsf{E}[D](1 - \mathsf{P}(\widetilde{T} < T)) + \mathsf{E}[\widetilde{T} - T]^+\},\end{aligned}$$

where we have used the fact that under periodic sampling, the queueing system evolution and the evolution of the statistical decision problem are independent, that is, $\widetilde{K}$ is independent of $\{D_1, D_2, \ldots\}$, and $\mathsf{E}[D]$ is the mean stationary queueing delay (of a batch). By writing $\mathsf{E}[D] = d(r)$ and using the fact that the problem $\min_{\Pi_\alpha} \mathsf{E}[\widetilde{T} - T]^+$ is solved by a policy $\pi_\alpha^* \in \Pi_\alpha$ with $\mathsf{P}(\widetilde{T} < T) = \alpha$, the problem becomes

$$d(r)(1 - \alpha) + \min_{\Pi_\alpha} \mathsf{E}[\widetilde{T} - T]^+ = (d(r) + l(r))(1 - \alpha) - \rho \cdot l(r) + \frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}[\widetilde{K} - K]^+,$$

where $l(r)$ is the delay due to sampling. Notice that $\min_{\Pi_\alpha} \mathsf{E}[\widetilde{K} - K]^+$ is the basic change detection problem at the sampling instants. □

Remark 3.4. It is important to note that the independence between $\widetilde{K}$ and $\{D_1, D_2, \ldots\}$ arises from periodic sampling. This is conditional independence, given the rate of the periodic sampling process. If, in general, one considers a model in which the sampling is at random times (e.g., the sampling process randomly alternates between periodic sampling at two different rates or if adaptive sampling is used), then we can view it as a time-varying sampling rate, and the asserted independence will not hold.

We conclude that the problem defined in Equation (2) effectively decouples into the sum of the optimal delay in the original optimal detection problem, that is, $\frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}[\widetilde{K} - K]^+$, as in Veeravalli [2001]; a part that captures the network delay, that is, $d(r)(1 - \alpha)$; and a part that captures the sampling delay, that is, $l(r)(1 - \alpha) - \rho l(r)$.

4. NETWORK DELAY MODEL

From Theorem 1, it is clear that in NODM processing, the optimal decision device and the queueing system are decoupled. Thus, one can employ an optimal sequential change detection procedure (see Shiryaev [1978]) for any random access network (in between the sensor nodes and the fusion center). Also, NODM is oblivious to the random access network (in between the sensor nodes and the fusion center) and processes a received batch as though it had just been generated. In the case of NADM (which we describe in Section 5), the decision maker processes samples taking network delays into account, thus requiring knowledge of network dynamics. In this section, we provide a simple model for the random access network that facilitates the analysis and optimization of NADM.

N sensors form a star topology[1] (see Figure 8) ad hoc wireless sensor network with the fusion center as the hub. They synchronously sample their environment at the rate of r samples periodically per slot. At sampling instant $t_b = b/r$, sensor node i generates a packet containing sample value $X_b^{(i)}$ (or some quantized version of it). This packet is then queued first-in-first-out in the buffer behind the radio link. It is as if each sample is a *fork* operation that puts a packet into each sensor queue (see Figure 6).

The sensor nodes contend for access to the radio channel and transmit packets when they succeed. The service is modeled as follows. As long as there are packets in any of the queues, successes are assumed to occur at the constant rate of σ ($0 < \sigma < 1$) per slot, with intervals between the successes being independent and identically distributed and geometrically distributed random variables with mean $1/\sigma$. If, at the time a success occurs, there are n nodes contending (i.e., n queues are nonempty), then the success is ascribed to any one of these n nodes, with equal probability.

The N packets corresponding to a sample arrive at random times at the fusion center. If the fusion center needs to accumulate all the N components of each sample, then it must wait for that component of every sample that is last to depart its mote. This is a *join* operation (see Figure 6).

It is easily recognized that our service model, in the case of NODM, is the discrete time equivalent of generalized processor sharing (GPS—see, for example, Kumar et al. [2004]), which can be called the FJQ-GPS (fork-join queue—(see Baccelli and Makowski [1990])—with GPS service). In the case of NADM, the service model is just the GPS.

In IEEE 802.11 and IEEE 802.15.4 if appropriate parameters are used, then the adaptive backoff mechanism can achieve a throughput that is roughly constant over a wide range of n, that is, the number of contending nodes. This is well known for CSMA/CA implementation in IEEE 802.11 wireless LANs; see, for example, Figure 9 [Kumar et al. 2008]. For each physical layer rate, the network service rate remains fairly constant with increasing number of nodes. From Figure 10 (taken from Singh et al. [2008]), it can be seen that with the default backoff parameters, the saturation throughput of a star topology IEEE 802.15.4 network decreases with the number of nodes N, but with the backoff multiplier $= 3$, the throughput remains almost constant from $N = 10$ to $N = 50$ [Singh et al. 2008]; thus, in the latter case, our GPS model can be applicable.

THEOREM 2. *The stationary delay D is a proper random variable with a finite mean if and only if $Nr < \sigma$.*

PROOF. See Appendix II. □

Thus, for the FJQ–GPS queueing system to be stable, sampling rate r is chosen such that $r < \frac{\sigma}{N}$.

5. NETWORK AWARE DECISION MAKING (NADM)

In Section 3, we formulated the problem of NODM quickest change detection over a random access network and showed that (when the decision instants are U_k, as shown in Figure 5) the optimal decision maker is independent of the random access network, under periodic sampling. Hence, the Shiryaev procedure, which is shown to

[1]Note that Theorem 1 is more general and does not assume a star topology.

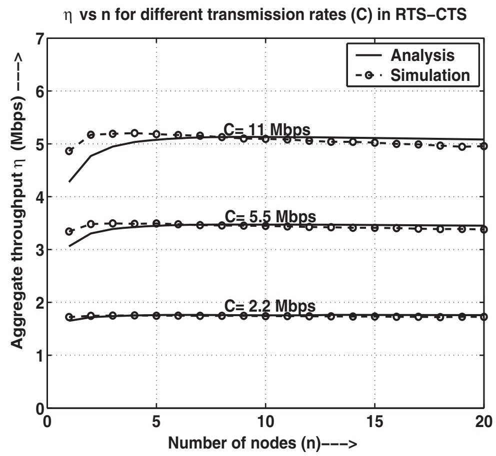

Fig. 9. The aggregate saturation throughput η of an IEEE 802.11 network plotted against the number of nodes in the network, for various physical layer bit rates: 2.2 Mbps, 5.5 Mbps, and 11 Mbps . The two curves in each plot correspond to an analysis and an NS–2 simulation.

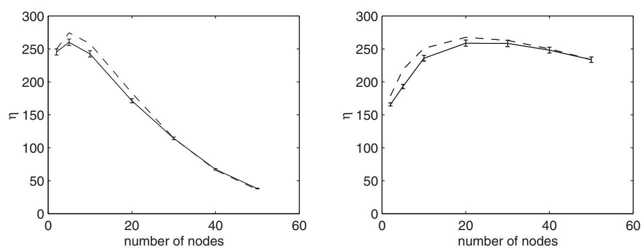

Fig. 10. The aggregate saturation throughput η of an IEEE 802.15.4 star topology network plotted against the number of nodes in the network. Throughput obtained with default backoff parameters is shown on the left and that obtained with backoff multiplier = 3 is shown on the right. The two curves in each plot correspond to an analysis and an NS–2 simulation.

be delay-optimal in the classical change detection problem (see Shiryaev [1978]), can be employed in the decision device independently of the random access network. It is to be noted that the decision maker in the NODM case waits for a complete batch of N samples to arrive to make a decision about the change. Thus, the mean detection delay of the NODM has a network-delay component corresponding to a batch of N samples. In this section, we provide an alternative mechanism of fusion at the decision device called Network Aware Decision Making (NADM), in which the fusion algorithm does not wait for an entire batch to arrive but processes the samples as soon as they arrive in a time-causal manner.

We now describe the processing in NADM. Whenever a node (successfully) transmits a sample across the random access network, it is delivered to the decision maker if the decision maker has received all the samples from all the batches generated earlier. Otherwise, the sample is an out-of-sequence sample and is queued in the sequencing buffer. It follows that whenever the (successfully) transmitted sample is the last component of the batch that the decision maker is looking for, then the *head of line* (HOL) components (if any) in the queues of the sequencing buffer are also delivered to the decision maker. This is because these HOL samples belong to the next batch that the decision maker should process. The decision maker makes a decision about the change at the beginning of every time slot, irrespective of whether it receives a sample or not.

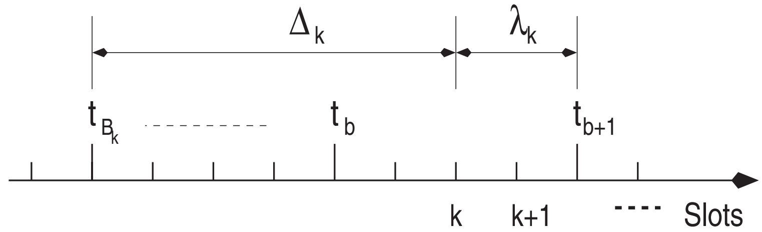

Fig. 11. At time k, the decision maker expects samples (or processes samples) from batch B_k. Also, at time k, λ_k is the number of slots to go for the next sampling instant, and Δ_k is the number of slots back, at which batch B_k is generated.

In NADM, whenever the decision maker receives a sample, it takes into account the network-delay of the sample while computing the decision statistic. The network-delay is a part of the state of the queueing system which is available to the decision maker. Thus, unlike NODM, the state of the queueing system also plays a role in decision making.

In Section 5.1, we define the state of the queueing system. In Section 5.2, we define the dynamical system whose change of state (from 0 to 1) is the subject of interest to us. We define the state of the dynamical system as a tuple that contains the queueing state, the state of nature, and a delayed state of nature. The delayed state of nature is included in the state of the system, so that the (delayed) sensor-observations that the decision maker receives at time instant $k+1$ depend only on the state, the control, and the noise of the system at time instant k—a property which is desirable to define a sufficient statistic (see page 244 at Bertsekas [2000a]). We explain the evolution of the state of the dynamical system in Section 5.3. In Section 5.4, we formulate the NADM change detection problem and find a sufficient statistic for the observations in Section 5.5. In Section 5.6, we provide the optimal decision rule for the NADM change detection problem.

5.1. Notation and State of the Queueing System

Recall the notation introduced in Section 2. Time progresses in slots, indexed by $k = 0, 1, 2 \ldots$; the beginning of slot k is the time instant k. Also, the time instant just after the beginning of time slot is denoted by $k+$.[2] Recall that the nodes take samples at instants $1/r, 2/r, 3/r, \ldots$. We define the state of the queueing system here. Note that the queueing system evolves over slots.

—$\lambda_k \in \{1, 2, \ldots, 1/r\}$ denotes the number of time slots to go for the next sampling instant at the beginning of time slot k (see Figure 11). Thus,

$$\lambda_k := \frac{1}{r} - \left(k \mod \frac{1}{r}\right). \tag{4}$$

Thus, $\lambda_0 = \frac{1}{r}, \lambda_1 = \frac{1}{r} - 1, \ldots$, and at the sampling instants t_b, $\lambda_{t_b} = \frac{1}{r}$.

—$B_k \in \{1, 2, 3, \ldots\}$ denotes the index of the batch that is expected to be or is being processed by the decision maker at the beginning of time slot k. Note $B_0 = B_1 = \cdots = B_{1/r} = 1$. Also, note that batch B_k is generated at time instant B_k/r.

[2]Note that the notation $t+$ denotes a time embedded to the right of t and is different from the notation $(x)^+$. Recall that $(x)^+ := \max\{x, 0\}$.

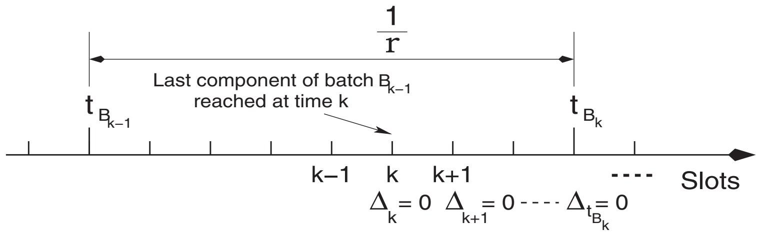

Fig. 12. Illustration of a scenario in which $\Delta_k = 0$. If the last component from batch B_{k-1} is received at k, and if there is no sampling instant between $t_{B_{k-1}}$ and k, then $\Delta_k = 0$. Also, note in this case that $\Delta_k = \Delta_{k+1} = \cdots = \Delta_{t_{B_k}} = 0$. In this scenario, at time instants $k, k+1, \ldots, t_{B_k}$, all the queues at the sensor nodes and the sequencer are empty, and at time instant $t_{B_k}+$, all sensor node queues have one packet which is generated at t_{B_k}.

—$\Delta_k \in \{0, 1, 2, \ldots\}$ denotes the delay in the number of time slots between time instants k and B_k/r (see Figure 11), such that

$$\Delta_k := \max\left\{k - \frac{B_k}{r}, 0\right\}. \tag{5}$$

Note that the batches of samples taken after B_k/r and up to (and including) k are queued either in the sensor node queues or in the sequencing buffer in the fusion center. If the fusion center receives a sample at time k, which is the last sample from batch B_{k-1}, then $B_k = B_{k-1} + 1$. If the sampling instant of the B_kth batch is later than k (i.e., $B_k/r > k$), then $\Delta_k = 0$ (up to time B_k/r, at which instant a new batch is generated). This corresponds to the case in which all the samples generated up to time slot k have already been processed by the decision maker (see Figure 12). In particular, $\Delta_0 = \Delta_1 = \cdots = \Delta_{\frac{1}{r}-1} = 0$.

—$L_k^{(i)} \in \{0, 1, 2, \ldots\}$ denotes the queue length of the ith sensor node just after the beginning of time slot k (i.e., at time instant $k+$). The vector of queue lengths is $\mathbf{L}_k = [L_k^{(1)}, L_k^{(2)}, \ldots, L_k^{(N)}]$. Let $N_k \in \{0, 1, 2, \ldots, N\}$ be the number of nonempty queues at the sensor nodes, just after the beginning of time slot k, given by

$$N_k := \sum_{i=1}^{N} \mathbf{1}_{\{L_k^{(i)} > 0\}},$$

that is, the number of sensor nodes contending for slot k is N_k. Hence, using the network model we have provided in Section 4, the evolution of $L_k^{(i)}$ (see Figure 13) is given by the following.

$$L_0^{(i)} = 0,$$
$$L_{k+1}^{(i)} = \begin{cases} L_k^{(i)} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}} & \text{w.p. } 1 \quad \text{if } N_k = 0, \\ L_k^{(i)} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}} & \text{w.p. } (1-\sigma) \quad \text{if } N_k > 0, \\ \max\{L_k^{(i)} - 1, 0\} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}} & \text{w.p. } \frac{\sigma}{N_k} \quad \text{if } N_k > 0. \end{cases}$$

Note that when all the samples generated up to time slot k have already been processed by the decision maker and k is not a sampling instant, that is, $\Delta_k = 0$ and $\lambda_k \neq 1/r$, then $\mathbf{L}_k = \mathbf{0}$ (as there are no outstanding samples in the system). For example, $\mathbf{L}_1 = \mathbf{L}_2 = \cdots = \mathbf{L}_{1/r-1} = \mathbf{0}$. Also, note that just after sampling instant t_b, $L_{t_b}^{(i)} \geqslant 1$.

—$M_k \in \{0, 1, 2, \ldots, N\}$ denotes the index of the node that successfully transmits in slot k. $M_k = 0$ means that there is no successful transmission in slot k. Thus, from the

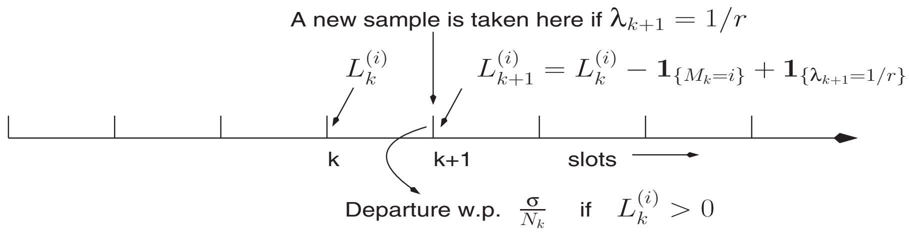

Fig. 13. The evolution of $L_k^{(i)}$ from time slot k to time slot $k+1$. If, during time slot k, node i (successfully) transmits a packet to the fusion center (i.e., $M_k = i$), then that packet is flushed out of its queue at the end of time slot k. Also, a new sample is generated (every $1/r$ slots) exactly at the beginning of a time slot. Thus, $L_{k+1}^{(i)}$, the queue length of sensor node i (just after the beginning of the time slot $k+1$ (i.e., at $(k+1)+$)) is given by $L_{k+1}^{(i)} = L_k^{(i)} - \mathbf{1}_{\{M_k=i\}} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}}$.

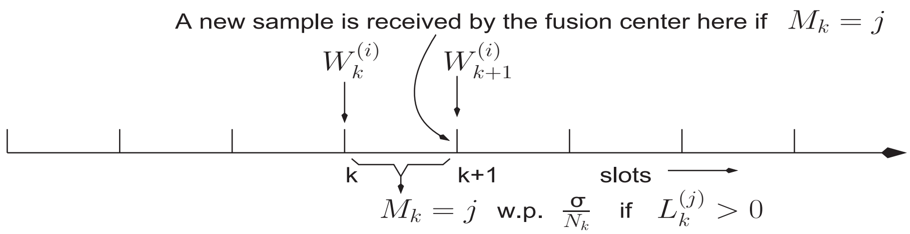

Fig. 14. The evolution of $W_k^{(i)}$ from time slot k to time slot $k+1$. If a sample from node i is transmitted (successfully) during time slot k (i.e., $M_k = i$), then it is received by the fusion center at the end of time slot k (i.e., at $(k+1)-$). If this sample is from batch B_k, it is passed on to the decision maker. Otherwise, it is queued in the sequencing buffer, in which case $W_{k+1}^{(i)} = W_k^{(i)} + 1$. On the other hand, if a sample from some other node j is transmitted (successfully) during time slot k (i.e., $M_k = j \neq i$), and if this sample is the last component to be received from batch B_k by the fusion center, then the HOL packet of the ith sequencing queue (if any) is also delivered to the decision maker. Thus, in this case, $W_{k+1}^{(i)} = \max\{W_k^{(i)} - 1, 0\}$. Note that $W_{k+1}^{(i)}$ refers to the queue length corresponding to node i at the sequencer at the beginning of time slot $k+1$.

network model provided in Section 4, we note that

$$M_k = \begin{cases} 0 \text{ w.p. } 1 & \text{if } N_k = 0, \\ 0 \text{ w.p. } (1-\sigma) & \text{if } N_k > 0, \\ j \text{ w.p. } \frac{\sigma}{N_k} & \text{if } L_k^{(j)} > 0, \; j = 1, 2, \ldots, N. \end{cases}$$

—$W_k^{(i)} \in \{0, 1, 2, \ldots\}$ denotes the queue length of the ith sequencing buffer at time k. The vector of queue lengths is given by $\mathbf{W}_k = \left[W_k^{(1)}, W_k^{(2)}, \ldots, W_k^{(N)}\right]$. Note that $\mathbf{W}_k = \mathbf{0}$ if $\Delta_k = 0$, that is, the sequencing buffer is empty if there are no outstanding samples in the system. In particular, $\mathbf{W}_0 = \mathbf{W}_1 = \cdots = \mathbf{W}_{\frac{1}{r}} = \mathbf{0}$. The evolution of $W_k^{(i)}$ is explained in Figure 14. If a sample from node i of a batch later than B_k is successfully transmitted during slot k, then $W_{k+1}^{(i)} = W_k^{(i)} + 1$. If a sample from node j of batch B_k is successfully transmitted, and if it is the last sample to be received from batch B_k, then the queue lengths of the sequencing buffer are decremented by 1, that is, $W_{k+1}^{(i)} = \max\{W_k^{(i)} - 1, 0\}$.

—$R_k^{(i)} \in \{0, 1\}$ denotes whether the sample $X_{B_k}^{(i)}$ has been received and processed by the decision maker, at time k. $R_k^{(i)} = 0$ means that the sample $X_{B_k}^{(i)}$ has not yet been received by the decision maker, and $R_k^{(i)} = 1$ means that the sample $X_{B_k}^{(i)}$ has been received and processed by the decision maker. The vector of $R_k^{(i)}$s is given by $\mathbf{R}_k = [R_k^{(1)}, R_k^{(2)}, \ldots, R_k^{(N)}]$. Note that if $R_k^{(i)} = 0$, then $W_k^{(i)} = 0$, that is, the ith sequencing buffer is empty if the sample expected by the decision maker has not yet been transmitted. Also note that when $\Delta_k = 0$, $\mathbf{R}_k = \mathbf{0}$, as the samples from the current batch B_k have yet to be generated or have just been generated.

We now relate the queue lengths $L_k^{(i)}$ and $W_k^{(i)}$. Note that at the beginning of time slot k, $\left\lfloor \frac{k}{1/r} \right\rfloor$ batches have been generated so far, of which $B_k - 1$ batches are completely received by the decision maker. In batch B_k, the ith sample is received by the decision maker, if $R_k^{(i)} = 1$. Hence, at time k, $B_k - 1 + R_k^{(i)}$ samples generated by node i have been processed by the decision maker and the remaining samples are in the sensor and sequencing queues. Thus, we have

$$\begin{aligned} L_k^{(i)} + W_k^{(i)} &= \left\lfloor \frac{k}{1/r} \right\rfloor - (B_k - 1) - R_k^{(i)} \\ &= \left\lfloor \frac{k - B_k/r + 1/r}{1/r} \right\rfloor - R_k^{(i)} \\ &= \begin{cases} \left\lfloor \frac{\Delta_k}{1/r} \right\rfloor + 1 - R_k^{(i)} & \text{if } k > B_k/r, \\ 1 - R_k^{(i)} & \text{if } k = B_k/r, \\ -R_k^{(i)} & \text{if } k < B_k/r. \end{cases} \end{aligned} \tag{6}$$

Recalling the definition of Δ_k, we write Equation (6) as

$$L_k^{(i)} + W_k^{(i)} = \begin{cases} \left\lfloor \frac{\Delta_k}{1/r} \right\rfloor + 1 - R_k^{(i)} & \text{if } \Delta_k > 0, \\ 1 & \text{if } \Delta_k = 0, \lambda_k = 1/r, \\ 0 & \text{if } \Delta_k = 0, \lambda_k \neq 1/r. \end{cases} \tag{7}$$

Note that in Equation (7) $\Delta_k = 0$, $\lambda_k = 1/r$ (or equivalently $k = B_k/r$) corresponds to the case when the samples of batch B_k have just been taken and all the samples from all previous batches have been processed. Thus, in this case, $L_k^{(i)} = 1$ (as $W_k^{(i)} = 0$). In the case of $\Delta_k = 0$, $\lambda_k \neq 1/r$ (or equivalently $k < B_k/r$), all the samples from all previous batches have been processed, and a new sample from batch B_k is not yet taken. Thus, in this case, $L_k^{(i)} = 0$ (and $W_k^{(i)} = 0$). Hence, given $\mathbf{Q}_k = \left[\lambda_k, B_k, \Delta_k, \mathbf{W}_k, \mathbf{R}_k\right]$, the queue lengths $L_k^{(i)}$s can be computed as

$$L_k^{(i)} = \phi_{L^{(i)}}(\mathbf{Q}_k) := \begin{cases} \left\lfloor \frac{\Delta_k}{1/r} \right\rfloor + 1 - R_k^{(i)} - W_k^{(i)} & \text{if } \Delta_k > 0, \\ 1 & \text{if } \Delta_k = 0, \lambda_k = 1/r, \\ 0 & \text{if } \Delta_k = 0, \lambda_k \neq 1/r. \end{cases} \tag{8}$$

$$\text{Also, } N_k = \phi_N(\mathbf{Q}_k) := \sum_{i=1}^{N} \mathbf{1}_{\{\phi_{L^{(i)}}(\mathbf{Q}_k) > 0\}}. \tag{9}$$

Thus, the state of the queueing system at time k can be expressed as $\mathbf{Q}_k = [\lambda_k, B_k, \Delta_k, \mathbf{W}_k, \mathbf{R}_k]$. Note that the decision maker can observe the state $\mathbf{Q}_k$ perfectly. The evolution of the queueing system is explained in Section 5.2.

5.2. Evolution of the Queueing System

The evolution of the queueing system from time k to time $k + 1$ depends only on M_k, that is the success/no-success of contention on the random access channel. Note that the evolution of λ_k is deterministic and that of Δ_k depends on B_k. Hence, to describe the evolution of $\mathbf{Q}_k$, it is enough to explain the evolution of B_k, $\mathbf{W}_k$, and $\mathbf{R}_k$ for various cases of M_k. Let $\mathbf{Y}_{k+1} \in \{\emptyset\} \cup (\cup_{n=1}^{N} \mathcal{X}^n)$ denote the vector of samples received (if any) by the decision maker at the beginning of slot $k+1$ (i.e., the decision maker can receive a vector of n samples, where n ranges from 0 to N).

At the beginning of time slot $k+1$, the following possibilities arise.

—*No successful transmission*. This corresponds to either the case i) in which all the queues are empty at the sensor nodes ($N_k = 0$) case or (ii) in which some queues are non–empty at the sensor nodes ($N_k > 0$), there are no queue attempts, or there is more than one attempt (resulting in a collision). In either case, $M_k = 0$ and the decision maker does not receive any sample, that is, $\mathbf{Y}_{k+1} = \emptyset$. In this case, it is clear that $B_{k+1} = B_k$, $\mathbf{W}_{k+1} = \mathbf{W}_k$, and $\mathbf{R}_{k+1} = \mathbf{R}_k$.

—*Successful transmission of node j's sample from a later batch*. This corresponds to the case when the decision maker has already received the jth component of the current batch B_k (i.e., $R_k^{(j)} = 1$) and has not received some sample, say $i \neq j$, from batch B_k (i.e., $R_k^{(i)} = 0$, for some i). The received sample (is an out-of-sequence sample and) is queued in the sequencing buffer ($W_{k+1}^{(j)} = W_k^{(j)} + 1$). Thus, in this case, $M_k = j$, and the decision maker does not receive any sample, that is, $\mathbf{Y}_{k+1} = \emptyset$. In this case, it is clear that $B_{k+1} = B_k$, $\mathbf{W}_{k+1} = \mathbf{W}_k + \mathbf{e}_j$, and $\mathbf{R}_{k+1} = \mathbf{R}_k$.

—*Successful transmission of node j's current sample, which is not the last component of batch B_k*. This corresponds to the case when the decision maker has not received the jth component of batch B_k before time slot k ($R_k^{(j)} = 0$), and it has received all the samples that are generated earlier than that of the successful sample. Also, the fusion center is yet to receive some other component of batch B_k (i.e., $\sum_{i=1}^{N} R_k^{(i)} < N - 1$). Thus, in this case, $M_k = j$, and the decision maker receives the sample $\mathbf{Y}_{k+1} = X_{B_k}^{(j)}$. In this case, it is clear that $B_{k+1} = B_k$, $\mathbf{W}_{k+1} = \mathbf{W}_k$, and $\mathbf{R}_{k+1} = \mathbf{R}_k + \mathbf{e}_j$.

—*Successful transmission of node j's current sample, which is the last component of the batch B_k*. This corresponds to the case when the decision maker has not received the jth component of batch B_k before time slot k ($R_k^{(j)} = 0$), and it has received all the samples that are generated earlier than that of the successful sample. Also, this sample is the last component of batch B_k that is received by the fusion center (i.e., $\sum_{i=1}^{N} R_k^{(i)} = N - 1$). In this case (along with the received sample), the queues of the sequencing buffer deliver the head of line (HOL) components (which correspond to the batch index B_k+1), if any, to the decision maker, and the queues are decremented by one ($W_{k+1}^{(i)} = \max\{W_k^{(i)} - 1, 0\}$). Thus, $M_k = j$ and the decision maker receive the vector of samples $\mathbf{Y}_{k+1} = [X_{B_k}^{(j)}, X_{B_k+1}^{(i'_1)}, X_{B_k+1}^{(i'_2)}, \dots, X_{B_k+1}^{(i'_{n-1})}]$, where $W_k^{(i)} > 0$ for $i \in \{i'_1, i'_2, \dots i'_{n-1}\}$, and $W_k^{(i)} = 0$ for $i \notin \{i'_1, i'_2, \dots i'_{n-1}\}$. In this case, $B_{k+1} = B_k + 1$, $\mathbf{W}_{k+1} = \mathbf{W}_k - \mathbf{e}_{i'_1} - \mathbf{e}_{i'_2} - \cdots - \mathbf{e}_{i'_{n-1}}$, and $\mathbf{R}_{k+1} = \mathbf{e}_{i'_1} + \mathbf{e}_{i'_2} + \cdots + \mathbf{e}_{i'_{n-1}}$.

Thus, the state of the queueing system at time $k+1$ can be described by

$$\begin{aligned} \mathbf{Q}_{k+1} &= \phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k) \\ &:= \left[\phi_\lambda(\mathbf{Q}_k, M_k), \phi_B(\mathbf{Q}_k, M_k), \phi_\Delta(\mathbf{Q}_k, M_k), \phi_{\mathbf{W}}(\mathbf{Q}_k, M_k), \phi_{\mathbf{R}}(\mathbf{Q}_k, M_k)\right]. \end{aligned}$$

In the next section, we provide a model of the dynamical system whose state has the state of nature Θ_k as one of its constituents. The quickest detection of change of Θ_k from 0 to 1 (at a random time T) is the focus of this article.

5.3. System State Evolution Model

Let $\Theta_k \in \{0, 1\}, k \geqslant 0$ be the state of nature at the beginning of time slot k. Recall that T is the change point, that is, for $k < T$, $\Theta_k = 0$ and for $k \geqslant T$, $\Theta_k = 1$, and that the distribution of T is given in Equation (1). The state Θ_k is observed only through the sensor measurements which are delayed. We will formulate the optimal NADM change detection problem as a partially observable Markov decision process (POMDP) with the delayed observations. The approach and the terminology used here is in accordance with the stochastic control framework in Bertsekas [2000a]. At time k, a sample (if any) that the decision maker receives is generated at time $B_k/r < k$ (i.e., samples arrive at the decision maker with a network-delay of $\Delta_k = k - \frac{B_k}{r}$ slots). To make an inference about Θ_k from the sensor measurements, we must consider the vector of states of nature that corresponds to the time instants $k - \Delta_k, k - \Delta_k + 1, \ldots, k$. We define the vector of states at time k as $\mathbf{\Theta}_k := [\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \ldots, \Theta_k]$. Note that the length of the vector depends on the network-delay Δ_k. When $\Delta_k > 0$, $\mathbf{\Theta}_k = [\Theta_{\frac{B_k}{r}}, \Theta_{\frac{B_k}{r}+1}, \ldots, \Theta_k]$, and when $\Delta_k = 0$, $\mathbf{\Theta}_k$ is just $[\Theta_k]$.

Consider the discrete-time system, which at the beginning of time slot k is described by the state

$$\Gamma_k = [\mathbf{Q}_k, \mathbf{\Theta}_k],$$

where we recall that

$$\begin{aligned}\mathbf{Q}_k &= [\lambda_k, B_k, \Delta_k, \mathbf{W}_k, \mathbf{R}_k],\\ \mathbf{\Theta}_k &= [\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \ldots, \Theta_k].\end{aligned}$$

Note that $\Gamma_0 = [[\frac{1}{r}, 1, 0, \mathbf{0}], \Theta_0]$. At each time slot k, we have the following set of controls $\{0, 1\}$, where 0 represents "take another sample", and 1 represents "stop and declare change." Thus, at time slot k, when the control chosen is 1, the state Γ_{k+1} is given by a terminal absorbing state t; when the control chosen is 0, the state evolution is given by $\Gamma_{k+1} = [\mathbf{Q}_{k+1}, \mathbf{\Theta}_{k+1}]$, where

$$\begin{aligned}\mathbf{Q}_{k+1} &= \boldsymbol{\phi}_{\mathbf{Q}}(\mathbf{Q}_k, M_k),\\ \mathbf{\Theta}_{k+1} &= \begin{cases}\left[\Theta_k + \mathbf{1}_{\{T=k+1\}}\right], & \text{if } \Delta_{k+1} = 0\\ \left[\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \ldots, \Theta_k, \Theta_k + \mathbf{1}_{\{T=k+1\}}\right], & \text{if } \Delta_{k+1} = \Delta_k + 1\\ \left[\Theta_{k-\Delta_k+\frac{1}{r}}, \Theta_{k-\Delta_k+\frac{1}{r}+1}, \ldots, \Theta_k, \Theta_k + \mathbf{1}_{\{T=k+1\}}\right], & \text{if } \Delta_{k+1} = \Delta_k + 1 - \frac{1}{r}.\end{cases}\\ &=: \boldsymbol{\phi}_{\mathbf{\Theta}}\left(\mathbf{\Theta}_k, \mathbf{Q}_k, M_k, \mathbf{1}_{\{T=k+1\}}\right),\end{aligned} \tag{10}$$

where it is easy to observe that $\Theta_k + \mathbf{1}_{\{T=k+1\}} = \Theta_{k+1}$. When $\Delta_{k+1} = \Delta_k + 1$, batch B_k is still in service, and hence, in addition to the current state $\Theta_{k+1} = \Theta_k + \mathbf{1}_{\{T=k+1\}}$, we need to keep the states $\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \ldots, \Theta_k$. Also, when $\Delta_{k+1} = \Delta_k + 1 - \frac{1}{r}$, the batch index is incremented, and hence, the vector of states that determines the distribution of the observations sampled at or after B_{k+1}/r and before $k+1$ is given by $[\Theta_{k-\Delta_k+\frac{1}{r}}, \Theta_{k-\Delta_k+\frac{1}{r}+1}, \ldots, \Theta_k, \Theta_k + \mathbf{1}_{\{T=k+1\}}]$.

Define $O_k := \mathbf{1}_{\{T=k+1\}}$, and define $\mathbf{N}_k := [M_k, O_k]$ as the state-noise during time slot k. The distribution of state-noise $\mathbf{N}_k$, given the state of the discrete-time system Γ_k, is given by $\mathsf{P}(M_k = m, O_k = o|\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}])$ and is the product of the distribution functions $\mathsf{P}(M_k = m|\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}])$ and $\mathsf{P}(O_k = o|\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}])$. These distribution functions are provided in Appendix III.

The problem is to detect the change in the state Θ_k as early as possible by sequentially observing the samples at the decision maker.

5.4. The NADM Change-Detection Problem

We now formulate the NADM change detection problem in which the observations from the sensor nodes are sent over a random access network to the fusion center, and the fusion center processes then the samples in the NADM mode.

In Section 5.3, we defined the state $\Gamma_k = [\mathbf{Q}_k, \mathbf{\Theta}_k]$ on which we formulate the NADM change detection problem as a POMDP. Recall that at the beginning of slot k, the decision maker receives a vector of sensor measurements $\mathbf{Y}_k$ and observes the state $\mathbf{Q}_k$ of the queueing system. Thus, at time k, $\mathbf{Z}_k = [\mathbf{Q}_k, \mathbf{Y}_k]$ is the observation of the decision maker about the state of the dynamical system Γ_k.

Let $A_k \in \{0, 1\}$ be the control (or action) chosen by the decision maker after having observed $\mathbf{Z}_k$ at k. Recall that 0 represents "take another sample" and 1 represents the action "stop and declare change." Let $\mathbf{I}_k = [\mathbf{Z}_{[0:k]}, A_{[0:k-1]}]$ be the information vector[3] that is available to the decision maker at the beginning of time slot k. Let τ be a stopping time with respect to the sequence of random variables $\mathbf{I}_1, \mathbf{I}_2, \ldots$. Note that $A_k = 0$ for $k < \tau$ and $A_k = 1$ for $k \geqslant \tau$. We are interested in obtaining a stopping time τ (with respect to the sequence $\mathbf{I}_1, \mathbf{I}_2, \ldots$) that minimizes the mean detection delay subject to a constraint on the probability of false alarm, that is,

$$\begin{aligned} \min \quad & \mathsf{E}\left[(\tau - T)^+\right], \\ \text{such that} \quad & \mathsf{P}(\tau < T) \leqslant \alpha. \end{aligned} \tag{11}$$

Note that in the case of NADM, at any time k, a decision about the change is made based on information $\mathbf{I}_k$ (which includes the batch index we are processing and the delays). Thus, in the case of NADM, false alarm is defined as the event $\{\tau < T\}$ and, hence, $\tau \geqslant T$ is not classified as a false alarm, even if it is due to pre-change measurements only. However, in the case of NODM, this is classified as a false alarm, as the decision about the change is based on the batches received until time k.

Let c be the cost per unit delay in detection. We are interested in obtaining a stopping time τ^* that minimizes the expected cost (Bayesian risk), given by

$$\begin{aligned} C(c, \tau^*) &= \min_\tau \mathsf{E}\left[\mathbf{1}_{\{\Theta_\tau = 0\}} + c \cdot (\tau - T)^+\right] \\ &= \min_\tau \mathsf{E}\left[\mathbf{1}_{\{\Theta_\tau = 0\}} + c \cdot \sum_{k=0}^{\tau-1} \mathbf{1}_{\{\Theta_k = 1\}}\right] \\ &= \min_\tau \mathsf{E}\left[g_\tau(\Gamma_\tau, A_\tau) + \sum_{k=0}^{\tau-1} g_k(\Gamma_k, A_k)\right] \\ &= \min_\tau \mathsf{E}\left[\sum_{k=0}^{\infty} g_k(\Gamma_k, A_k)\right], \end{aligned} \tag{12}$$

where $\Gamma_k = [\mathbf{Q}_k, \mathbf{\Theta}_k]$ as defined earlier. Let $\boldsymbol{\theta} = [\theta_\delta, \theta_{\delta-1}, \ldots, \theta_1, \theta_0]$. We define for $k \leqslant \tau$ that

$$g_k([\mathbf{q}, \boldsymbol{\theta}], a) = \begin{cases} 0, & \text{if } \theta_0 = 0, a = 0 \\ 1, & \text{if } \theta_0 = 0, a = 1 \\ c, & \text{if } \theta_0 = 1, a = 0 \\ 0, & \text{if } \theta_0 = 1, a = 1, \end{cases} \tag{13}$$

and for $k > \tau$, $g_k(\cdot, \cdot) := 0$. Recall that $A_k = 0$ for $k < \tau$ and $A_k = 1$ for $k \geqslant \tau$. Note that A_k, the control at time slot k, depends only on $\mathbf{I}_k$. Thus, every stopping time τ

[3]The notation $\mathbf{Z}_{[k_1:k_2]} := \mathbf{Z}_{k_1}, \mathbf{Z}_{k_1+1}, \ldots, \mathbf{Z}_{k_2}$.

corresponds to a policy $\mu = (\mu_0, \mu_1, \ldots)$, such that $A_k = \mu_k(\mathbf{I}_k)$, with $A_k = 0$ for $k < \tau$ and $A_k = 1$ for $k \geqslant \tau$. Thus, Equation (12) can be written as

$$\begin{aligned} C(c, \tau^*) &= \min_{\mu} \mathsf{E}\left[\sum_{k=0}^{\infty} g_k(\Gamma_k, A_k)\right] \\ &= \min_{\mu} \sum_{k=0}^{\infty} \mathsf{E}\big[g_k(\Gamma_k, A_k)\big] \text{ (by monotone convergence theorem)} \\ &= \min_{\mu} \sum_{k=0}^{\infty} \mathsf{E}\big[g_k(\Gamma_k, \mu_k(\mathbf{I}_k))\big]. \end{aligned} \tag{14}$$

Since $\mathbf{\Theta}_k$ is observed only through $\mathbf{I}_k$, we look at a sufficient statistic for $\mathbf{I}_k$ in the next section.

5.5. Sufficient Statistic

In Section 5.2, we have illustrated the evolution of the queueing system $\mathbf{Q}_k$ and have shown, in different scenarios, the vector $\mathbf{Y}_k$ received by the decision maker. Recall from Section 5.2 that

$$\mathbf{Y}_{k+1} = \begin{cases} \emptyset, & \text{if } M_k = 0, \\ \emptyset, & \text{if } M_k = j > 0, R_k^{(j)} = 1, \\ Y_{k+1,0}, & \text{if } M_k = j > 0, R_k^{(j)} = 0, \sum_{i=1}^{N} R_k^{(i)} < N-1 \\ \big[Y_{k+1,0}, Y_{k+1,1}, \ldots, Y_{k+1,n}\big], & \text{if } M_k = j > 0, R_k^{(j)} = 0, \sum_{i=1}^{N} R_k^{(i)} = N-1, \\ & \sum_{i=1}^{N} \mathbf{1}_{\{W_k^{(i)} > 0\}} = n. \end{cases}$$

Note that $Y_{k+1,0}$ corresponds to $X_{B_k}^{(M_k)}$. The last part of the preceding equation corresponds to the last pending sample of batch B_k arriving at the decision maker at time $k+1$, with some samples from batch $B_k + 1$ $(= B_{k+1})$ also being released by the sequencer. In this case, the state of nature at the sampling instant of batch $B_{k+1} = B_k + 1$ is $\Theta_{k-\Delta_k+1/r}$. Note that $\Theta_{k-\Delta_k+1/r}$ is a component of the vector $\mathbf{\Theta}_k$, as $k - \Delta_k + 1/r = (B_k + 1)/r < k$. Thus, the distribution of $Y_{k+1,0}, Y_{k+1,1}, \ldots, Y_{k+1,n}$ is given by

$$f_{Y_{k+1,0}}(\cdot) = \begin{cases} f_0(\cdot), & \text{if } \Theta_{k-\Delta_k} = 0 \\ f_1(\cdot), & \text{if } \Theta_{k-\Delta_k} = 1 \text{ and} \end{cases}$$

$$f_{Y_{k+1,i}}(\cdot) = \begin{cases} f_0(\cdot), & \text{if } \Theta_{k-\Delta_k+1/r} = 0 \\ f_1(\cdot), & \text{if } \Theta_{k-\Delta_k+1/r} = 1. \end{cases}, \quad i = 1, 2, \ldots, n.$$

Thus, at time $k+1$, the current observation $\mathbf{Y}_{k+1}$ depends only on the previous state Γ_k, previous action A_k, and the previous noise of the system $\mathbf{N}_k$. Thus, a sufficient statistic is $[\mathsf{P}(\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] | \mathbf{I}_k)]_{[\mathbf{q}, \boldsymbol{\theta}] \in \mathcal{S}}$ (see page 244 at Bertsekas [2000a]), where $\mathcal{S}$ is the set of all states of the dynamical system defined in Sec. 5.3. Let $\mathbf{q} = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]$. Note that

$$\begin{aligned}
&\mathsf{P}\left(\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_k\right) \\
&= \mathsf{P}\left(\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k, \mathbf{Y}_k\right) \\
&= \mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left(\boldsymbol{\Theta}_k = \boldsymbol{\theta} \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\
&= \mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \\
&\quad \cdot \mathsf{P}\left([\Theta_{k-\delta}, \Theta_{k-\delta+1}, \ldots, \Theta_{k-1}, \Theta_k] = [\theta_\delta, \theta_{\delta-1}, \ldots, \theta_1, \theta_0] \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\
&= \mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left(\Theta_{k-\delta} = \theta_\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\
&\quad \cdot \prod_{j=1}^{\delta} \mathsf{P}\left(\Theta_{k-\delta+j} = \theta_{\delta-j} \,\middle|\, \Theta_{k-\delta+j'} = \theta_{\delta-j'}, j' = 0, 1, \ldots, j-1, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right)
\end{aligned} \tag{15}$$

Observe that

$$\begin{aligned}
&\mathsf{P}\left(\Theta_{k-\delta+j} = \theta_{\delta-j} \,\middle|\, \Theta_{[k-\delta:k-\delta+j-2]}, \Theta_{k-\delta+j-1} = 0, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\
&\quad = \begin{cases} 1-p, & \text{if } \theta_{\delta-j} = 0 \\ p, & \text{if } \theta_{\delta-j} = 1, \end{cases}
\end{aligned}$$

and

$$\begin{aligned}
&\mathsf{P}\left(\Theta_{k-\delta+j} = \theta_{\delta-j} \,\middle|\, \Theta_{[k-\delta:k-\delta+j-2]}, \Theta_{k-\delta+j-1} = 1, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\
&\quad = \begin{cases} 0, & \text{if } \theta_{\delta-j} = 0 \\ 1, & \text{if } \theta_{\delta-j} = 1. \end{cases}
\end{aligned}$$

This is because given $\Theta_{k-\delta}$, the events $\{\Theta_{k-\delta+j} = \theta_{\delta-j}\}$, $\left\{\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\}$ are conditionally independent. Thus, Equation (15) can be written as

$$\begin{aligned}
&\mathsf{P}\left(\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}\ ]\right) \big| \mathbf{I}_k \\
&= \begin{cases}
\mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left(\Theta_{k-\delta} = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right), & \text{if } \boldsymbol{\theta}\ = \mathbf{1} \\
\mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left(\Theta_{k-\delta} = 0 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \cdot (1-p)^{\delta-j-1} p, & \text{if } \boldsymbol{\theta} = [0, \ldots, 0, \underbrace{1}_{\theta_j}, \ldots, 1] \\
\mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left(\Theta_{k-\delta} = 0 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \cdot (1-p)^{\delta}, & \text{if } \boldsymbol{\theta} = \mathbf{0}.
\end{cases},
\end{aligned} \tag{16}$$

Define $\widetilde{\Theta}_k := \Theta_{k-\Delta_k}$, and define

$$\begin{aligned}
\Psi_k &:= \mathsf{P}\left(\widetilde{\Theta}_k = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right) \\
&= \mathsf{P}\left(\Theta_{k-\delta} = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right) \\
\Pi_k &:= \mathsf{P}\left(\Theta_k = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right) \\
&= \mathsf{P}\left(T \leqslant k \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right).
\end{aligned} \tag{17}$$

Thus, Equation (16) can be written as

$$\begin{aligned}
&\mathsf{P}\left(\Gamma_k = \left[[\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \boldsymbol{\theta}\right] \,\middle|\, \mathbf{I}_k\right) \\
&= \begin{cases}
\mathbf{1}_{\{\mathbf{Q}_k=[\lambda,b,\delta,\mathbf{w},\mathbf{r}]\}} \cdot \Psi_k, & \text{if } \boldsymbol{\theta} = \mathbf{1} \\
\mathbf{1}_{\{\mathbf{Q}_k=[\lambda,b,\delta,\mathbf{w},\mathbf{r}]\}} \cdot (1-\Psi_k) \cdot (1-p)^{\delta-j-1} p, & \text{if } \boldsymbol{\theta} = [0, \ldots, 0, \underbrace{1}_{\theta_j}, \ldots, 1] \\
\mathbf{1}_{\{\mathbf{Q}_k=[\lambda,b,\delta,\mathbf{w},\mathbf{r}]\}} \cdot (1-\Psi_k) \cdot (1-p)^{\delta}, & \text{if } \boldsymbol{\theta} = \mathbf{0}.
\end{cases}
\end{aligned} \tag{18}$$

We now find a relation between Π_k and Ψ_k in the following Lemma.

LEMMA 5.1. *The relation between the conditional probabilities Π_k and Ψ_k is given by*

$$\Pi_k \;=\; \Psi_k + (1-\Psi_k)\left(1-(1-p)^{\delta}\right). \tag{19}$$

PROOF. See Appendix IV. □

From Equation (18) and Lemma 5.1, it is clear that a sufficient statistic for $\mathbf{I}_k$ is $\nu_k = [\mathbf{Q}_k, \Pi_k]$. Also, we show in Appendix V that ν_k can be computed recursively, that is, when $A_k = 0$, $\nu_{k+1} = [\mathbf{Q}_{k+1}, \Pi_{k+1}] = [\mathbf{Q}_{k+1}, \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})]$, and when $A_k = 1$, $\nu_{k+1} = \mathsf{t}$, a terminal state. Thus, ν_k can be thought of as entering into a terminating (absorbing) state t at τ (i.e., $\nu_k = [\mathbf{Q}_k, \Pi_k]$, for $k < \tau$ and $\nu_k = \mathsf{t}$ for $k \geqslant \tau$). Since ν_k is sufficient, for every policy μ_k there corresponds a policy $\widetilde{\mu}_k$ such that $\mu_k(\mathbf{I}_k) = \widetilde{\mu}_k(\nu_k)$ (see page 244 at Bertsekas [2000a]).

5.6. Optimal Stopping Time τ

Let $\mathcal{Q}$ be the set of all possible states of the queueing system $\mathbf{Q}_k$. Thus, the state space of the sufficient statistic is $\mathcal{N} = (\mathcal{Q} \times [0,\ 1]) \cup \{\mathsf{t}\}$. Recall that the action space is $\mathcal{A} = \{0, 1\}$. Define the one-stage cost function $\widetilde{g} : \mathcal{N} \times \mathcal{A} \to \mathbb{R}_+$ as follows. Let $\nu \in \mathcal{N}$ be a state of the system, and let $a \in \mathcal{A}$ be a control. Then,

$$\widetilde{g}(\nu, a) \;=\; \begin{cases} 0 & \text{if } \nu = \mathsf{t} \\ c \cdot \pi & \text{if } \nu = [\mathbf{q}, \pi], a = 0 \\ 1-\pi & \text{if } \nu = [\mathbf{q}, \pi], a = 1. \end{cases}$$

Note from Equation (13) for $k \leqslant \tau$, that

$$\begin{aligned} \mathsf{E}\big[g_k(\Theta_k, A_k)\big] \;&=\; \mathsf{E}\big[g_k(\Theta_k, \mu_k(\mathbf{I}_k))\big] \\ &=\; \mathsf{E}\big[\mathsf{E}\big[g_k(\Theta_k, \mu_k(\mathbf{I}_k))\,\big|\,\mathbf{I}_k\big]\big] \\ &=\; \mathsf{E}\big[\widetilde{g}(\nu_k, \widetilde{\mu}_k(\nu_k))\big], \end{aligned}$$

and for $k > \tau$, that

$$\begin{aligned} \mathsf{E}\big[g_k(\Theta_k, A_k)\big] \;&=\; 0 \\ &=\; \mathsf{E}\big[\widetilde{g}(\mathsf{t}, \cdot)\big]. \end{aligned}$$

Since $\{\nu_k\}$ is a controlled Markov process, and the one-stage cost function $\widetilde{g}(\cdot,\cdot)$, the transition probability kernel for $A_k = 1$ and for $A_k = 0$ (i.e., $\mathsf{P}(\mathbf{Z}_{k+1}|\nu_k)$) does not depend on time k, and since the optimization problem defined in Equation (14) is over infinite horizon, it is sufficient to look for an optimal policy in the space of stationary Markov policies (see page 83 at Bertsekas [2000b]). Thus, the optimization problem defined in Equation (14) can be written as

$$\begin{aligned} C(c, \tau^*) \;&=\; \min_{\widetilde{\mu}} \sum_{k=0}^{\infty} \mathsf{E}\left[\widetilde{g}\big(\nu_k, \widetilde{\mu}_k(\nu_k)\big)\right] \\ &=\; \sum_{k=0}^{\infty} \mathsf{E}\left[\widetilde{g}\big(\nu_k, \widetilde{\mu}^*(\nu_k)\big)\right]. \end{aligned} \tag{20}$$

Thus, the optimal total cost is given by

$$J^*([\mathbf{q}_0, \pi_0]) \;=\; \sum_{k=0}^{\infty} \mathsf{E}[\widetilde{g}(\nu_k, \widetilde{\mu}^*(\nu_k))|\nu_0 = [\mathbf{q}_0, \pi_0]]. \tag{21}$$

The solution to the preceding problem is obtained by following Bellman's equation, which is given by

$$J^*([\mathbf{q}, \pi]) := \min\{1 - \pi, c\pi + \mathsf{E}[J^*(\mathbf{Q}_{k+1}, \phi_\Pi(\nu_k, \mathbf{Z}_{k+1}))|\nu_k = [\mathbf{q}, \pi]]\}, \tag{22}$$

where the function $\phi_\Pi(\nu_k, \mathbf{Z}_{k+1})$ is provided in Appendix V.

Remark 5.2. The optimal stationary Markov policy (i.e., the optimum stopping rule τ) in general depends on $\mathbf{Q}$. Hence, the decision delay and the queueing delay are coupled, unlike in the NODM case.

We characterize the optimal policy in the following theorem.

THEOREM 3. *The optimal stopping rule τ^* is a network-state dependent threshold rule on the a posteriori probability Π_k, that is, there exist thresholds $\gamma(\mathbf{q})$, such that*

$$\tau = \inf\{k \geqslant 0 : \Pi_k \geqslant \gamma(\mathbf{Q}_k)\}. \tag{23}$$

PROOF. See Appendix VI. □

In general, the thresholds $\gamma(\mathbf{Q}_k)$s (i.e., optimum policy) can be numerically obtained by solving Equation (22) using value iteration method (see pp. 88–90 at Bertsekas [2000b]). However, computing the optimal policy for the NADM procedure is hard, as the state space is huge even for moderate values of N. Hence, we resort to a suboptimal policy based on the following threshold rule motivated by the structure of the optimal policy.

$$\tau = \inf\{k \geqslant 0 : \Pi_k \geqslant \gamma\}, \tag{24}$$

where γ is chosen such that $\mathsf{P}(\tau < T) = \alpha$ is met.

Thus, we have formulated a sequential change detection problem in which the sensor observations are sent to the decision maker over a random access network, and the fusion center processes the samples in the NADM mode. The information for decision making now needs to include the network state $\mathbf{Q}_k$ (in addition to the samples received by the decision maker); we have shown that $[\mathbf{Q}_k, \Pi_k]$ is sufficient for the information history $\mathbf{I}_k$. Also, we have provided the structure for the optimal policy. Since obtaining the optimal policy is computationally hard, we gave a simple threshold-based policy, which is motivated by the structure of the optimal policy.

6. NUMERICAL RESULTS

Minimizing the mean detection delay not only requires an optimal decision rule at the fusion center but also involves choosing the optimal values of the sampling rate r and the number of sensors N. To explore this, we obtain the minimum decision delay for each value of the sampling rate r numerically and obtain the network delay via simulation.

6.1. Optimal Sampling Rate

Consider a sensor network with N nodes. For a given probability of false alarm, the decision delay (detection delay without the network-delay component) decreases with increase in sampling rate. This is due to the increase in the number of samples that the fusion center receives within a given time. But, as the sampling rate increases, the network delay increases due to the increased packet communication load in the network. Therefore, it is natural to expect the existence of a sampling rate r^* with $r^* < \sigma/N$, (the sampling rate should be less than σ/N for the queues to be stable; see Theorem 2) that optimizes the tradeoff between these two components of detection delay. Such an r^*, in the case of NODM, can be obtained by minimizing the following expression over r (recall Theorem 1).

$$(d(r)+l(r))(1-\alpha)-\rho\cdot l(r)+\frac{1}{r}\min_{\Pi_\alpha}\mathsf{E}\big[\widetilde{K}-K\big]^{+}.$$

Note that in the preceding expression, the delay term $\min_{\Pi_\alpha}\mathsf{E}[\widetilde{K}-K]^{+}$ also depends on the sampling rate r, via the probability of change $p_r = 1-(1-p)^{(1/r)}$. The delay due to coarse sampling $l(r)(1-\alpha)-\rho\cdot l(r)$ can be found analytically (see Appendix I). We can approximate the delay $\min_{\Pi_\alpha}\mathsf{E}[\widetilde{K}-K]^{+}$ by the asymptotic (as $\alpha \to 0$) delay as $\frac{|\ln(\alpha)|}{NI(f_1,f_0)+|\ln(1-p_r)|}$, where $I(f_1, f_0)$ is the Kullback-Leibler (KL) divergence between the pdfs f_1 and f_0 [Tartakovsky and Veeravalli 2005]. But obtaining the network-delay (i.e., $d(r)(1-\alpha)$) analytically is hard, and hence, an analytical characterisation of r^* appears intractable. Therefore, we have resorted to numerical evaluation.

The distribution of sensor observations are taken to be $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$,[4] before and after the change, respectively, for all the ten nodes. We choose the probability of occurrence of change in a slot to be $p = 0.0005$, that is, the mean time until change is 2,000 slots. $\min_{\Pi_\alpha}\mathsf{E}\big[\widetilde{K}-K\big]^{+}$ and $d(r)$ are obtained from simulations for $\alpha = 0.01$ and $\sigma = 0.3636$, and the expression for mean detection delay is plotted against r in Figure 15. Note that both NODM and NADM are threshold-based, and we obtain the corresponding thresholds for a target $\mathsf{P}_{\mathsf{FA}} = 0.01$ by simulation. These thresholds are then used to obtain the mean detection delay by simulation. In Figure 15, we also plot the approximate mean detection delay which is obtained through the expression for $l(r)$ and the approximation, $\min_{\Pi_\alpha}\mathsf{E}\big[\widetilde{K}-K\big]^{+} \approx \frac{|\ln(\alpha)|}{NI(f_1,f_0)+|\ln(1-p_r)|}$. We study this approximation as it provides an (approximate) explicit expression for the mean decision delay. The delay in the FJQ–GPS does not have a closed-form expression. Hence, we still need simulation for the delay, due to queueing network. It is to be noted that at $k = 0$, the size of all the queues is set to 0. The mean detection delay due to the procedure defined in Equation (24) is also plotted in Figure 15.

As would have been expected, we see from Figure 15 that the NADM procedure has a better mean detection delay performance than the NODM procedure. Note that $\sigma/N = 0.03636$, and hence for the queues to be stable (see Theorem 2), the sampling rate has to be less than $\sigma/N = 0.03636$ ($1/28 < 0.03636 < 1/27$). As the sampling rate r increases to 1/28 (the maximum allowed sampling rate), the queueing delay increases rapidly. This is evident from Figure 15. Also, we see from Figure 15 that operating at a sampling rate around $1/34 (\approx 0.0294)$ samples/slot would be optimal. The optimal sampling rate, is found to be approximately the same for NODM and NADM. At the optimal sampling rate, the mean detection delay of NODM is 90 slots, and that of NADM is 73 slots.

6.2. Optimal Number of Sensor Nodes (Fixed Observation Rate)

Now let us consider fixing $N \times r$. This is the number of observations the fusion center receives per slot in a network with N nodes sampling at a rate r (samples per slot). It is also a measure of the energy spent by the network per slot. Since it has been assumed that the observations are conditionally independent and identically distributed across the sensors and over time, it is natural to ask how beneficial it is to have more nodes sampling at a lower rate compared to fewer nodes sampling at a higher rate, when the number of observations per slot is the same. With $p = 0.0005$, $\alpha = 0.01$, and $\sigma = 0.3636$, and $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$, we present simulation results for two examples,

[4]As usual, $\mathcal{N}(a, v)$ denotes a normal distribution with mean a and variance v

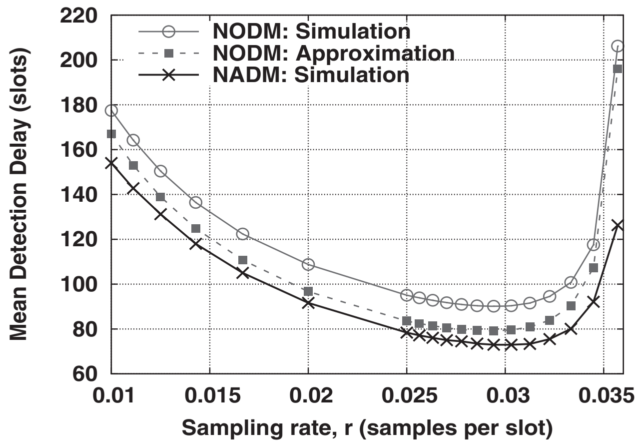

Fig. 15. Mean detection delay for $N = 10$ nodes is plotted against the sampling rate r for both NODM and NADM (defined in Equation (24)). For NODM, an approximate analysis is also plotted. This was obtained with the prior probability $\rho = 0$, $p = 0.0005$, probability of false alarm target $\alpha = 0.01$, $\sigma = 0.3636$, and with the sensor observations being $\mathcal{N}(0, 1)$ and $\mathcal{N}(1, 1)$, before and after the change, respectively.

the first one being $Nr = 1/3$ (the case of a heavily loaded network) and the second one being $Nr = 1/100$ (the case of a lightly loaded network, $Nr \ll \sigma$).

Figure 16 shows the plot of mean decision delay $l(r)(1 - \alpha - \rho) + \frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}[\widetilde{K} - K]^+$ versus the number of sensors when $Nr = 1/3$. As N increases, the sampling rate $r = 1/(3N)$ decreases, and hence, the coarse sampling delay $l(r)(1 - \alpha)$ increases; this can be seem to be approximately linear by analysis of the expression for $l(r)$, given in Appendix I. Also, as N increases, the decision maker gets more samples at the decision instants, hence, the delay due to the decision maker $\frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}[\widetilde{K} - K]^+$ decreases (this is evident from the right side of Figure 16). Figure 16 shows that in the region where N is large (i.e., $N \geqslant 20$) or N is very small (i.e., $N < 5$), as N increases, the mean decision delay increases. This is because as N increases in this region, the decrease in delay due to the decision maker is smaller compared to the increase in delay due to coarse sampling. However, in the region where N is moderate (i.e., for $5 \leqslant N < 20$), as N increases, the decrease in delay due to the decision maker is large compared to the increase in delay due to coarse sampling. Hence, in this region, the mean decision delay decreases with N. Therefore, we infer that when $N \times r = \frac{1}{3}$, deploying 20 nodes sampling at 1/60 samples per slot is optimal, when there is no network delay.

Figure 17 shows the mean detection delay (i.e., the network delay plus the decision delay shown in Figure 16) versus the number of nodes N for a fixed $N \times r = 1/3$. As the the number of nodes N increases, the sampling rate $r = 1/(3N)$ decreases. For large N (and equivalently small r), as in the case of NODM with the Shiryaev procedure, the network delay $d(r) \approx \frac{N}{\sigma}$ as it requires N (independent) successes, (each with probability σ) in the random access network to transport a batch of N samples (also, since the sampling rate r is small, one would expect that a batch is delivered before a new batch is generated), and the decision maker requires just one batch of N samples to stop (after the change occurs). Hence, for large N, the detection delay is $\approx l(r)(1 - \alpha) + d(r)(1 - \alpha) \approx l(r)(1 - \alpha) + \frac{N}{\sigma}(1 - \alpha)$. It is to be noted that for large N, to achieve a false alarm probability of α, the decision maker requires $N_\alpha < N$ samples. (The mean of the log-likelihood ratio (LLR) of received samples, after change, is the KL divergence between pdfs f_1 and f_0, given by $I(f_1, f_0) > 0$. Hence, the posterior probability—which is a function of LLR—increases with the the number of received

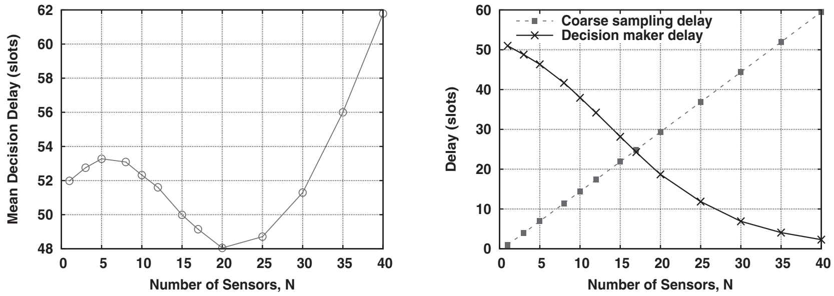

Fig. 16. Mean decision delay of NODM procedure for $N \times r = 1/3$ is plotted against the number of nodes N. The plot is obtained with $\rho = 0,\ p = 0.0005,\ \alpha = 0.01$, and with the sensor observations $\mathcal{N}(0, 1)$ and $\mathcal{N}(1, 1)$, before and after the change, respectively. The components of the mean decision delay, that is, the coarse sampling delay $(1 - \alpha)l(r) - \rho l(r)$ and the decision maker delay $\frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}[\widetilde{K} - K]^+$ are shown on the right.

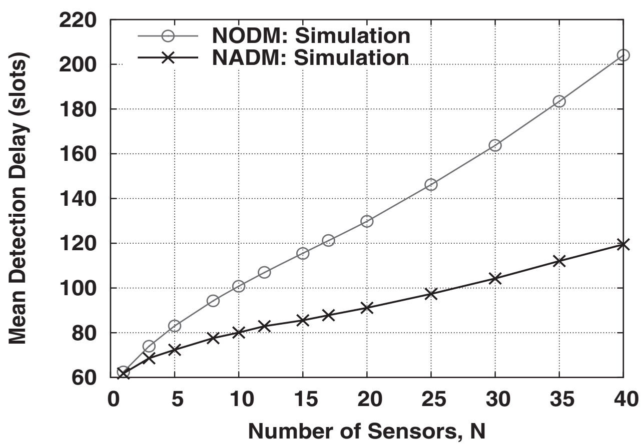

Fig. 17. Mean detection delay for $N \times r = 1/3$ is plotted against the number of nodes N. This was obtained with $\rho = 0,\ p = 0.0005,\ \alpha = 0.01\ \sigma = 0.3636$, and with sensor observations $\mathcal{N}(0, 1)$ and $\mathcal{N}(1, 1)$, before and after the change, respectively.

samples. Thus, to cross a threshold of $\gamma(\alpha)$, we need N_α samples). Thus, for large N, in the NADM procedure, the detection delay is approximately $l(r)(1 - \alpha) + \frac{N_\alpha}{\sigma}(1 - \alpha)$, where N_α/σ is the mean network-delay to transport N_α samples. Thus, for large N, the difference in the mean detection delay between NODM and NADM procedures is approximately $\frac{1-\alpha}{\sigma}(N - N_\alpha)$. Note that N_α depends only on α, and hence, the quantity $\frac{1-\alpha}{\sigma}(N - N_\alpha)$ increases with N. This behaviour is in agreement with Figure 17. Also, as $N \times r = 1/3$, we expect the network delay to be very large (as 1/3 is close to $\sigma = 0.3636$), and hence, having a single node is optimal, which is also evident from Figure 17.

It is also possible to find an example where the optimal number of nodes is greater than 1. For example, this occurs in the previous setting for $N \times r = 0.01$ (see Figure 18). Note that having $N = 10$ sensors is optimal for the NADM procedure. The NODM procedure makes the decision only when it receives a batch of N samples corresponding to a sampling instant, whereas NADM procedure makes the decision at every time slot, irrespective of whether it receives a sample in that time slot or not. Thus, the Bayesian update that NADM does at every time slot makes it stop earlier than NODM.

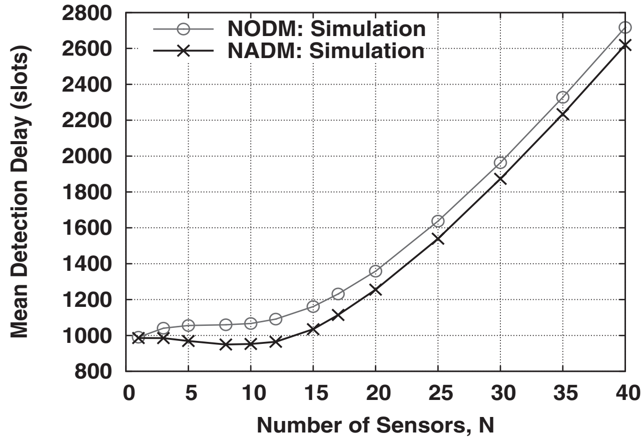

Fig. 18. Mean detection delay for $N \times r = 0.01$ is plotted against the the number of nodes N. This was obtained with $\rho = 0$, $p = 0.0005$, $\alpha = 0.01$, and with sensor observations $\mathcal{N}(0, 1)$ and $\mathcal{N}(1, 1)$, before and after the change, respectively.

7. CONCLUSIONS

In this work, we have considered the problem of minimizing the mean detection delay in an event detection on a small extent ad hoc wireless sensor network. We provide two ways of processing samples in the fusion center: (i) Network Oblivious (NODM) processing and (ii) Network Aware (NADM) processing. We show that in the NODM processing, under periodic sampling, the detection delay decouples into decision and network delays. An important implication of this is that an optimal sequential change detection algorithm can be used in the decision device, independently of the random access network. We also formulate and solve the change detection problem in the NADM setting, in which case the optimal decision maker needs to use the network state in its optimal stopping rule. Also, we study the network delay involved in this problem and show that it is important to operate at a particular sampling rate to achieve the minimum detection delay.

APPENDIXES

APPENDIX I

PROOF. (THEOREM 1).

$$\begin{aligned}
\min_{\Pi_\alpha} \mathsf{E}\big[(\widetilde{U} - T) I_{\{\widetilde{T} \geqslant T\}}\big] &= \min_{\Pi_\alpha} \mathsf{E}\left[\left(\widetilde{U} - \widetilde{T} + \widetilde{T} - \frac{K}{r} + \frac{K}{r} - T\right) I_{\{\widetilde{T} \geqslant T\}}\right] \\
&= \min_{\Pi_\alpha} \left\{ \mathsf{E}\big[(\widetilde{U} - \widetilde{T}) I_{\{\widetilde{T} \geqslant T\}}\big] + \mathsf{E}\left[\left(\frac{K}{r} - T\right) I_{\{\widetilde{T} \geqslant T\}}\right] \right. \\
&\quad \left. + \frac{1}{r} \mathsf{E}\big[(\widetilde{K} - K) I_{\{\widetilde{T} \geqslant T\}}\big] \right\}. \qquad (25)
\end{aligned}$$

Note that in Equation (25), the first term is the queueing delay, the second term is the coarse sampling delay, and the third term is the decision delay (all delays being in slots). Consider the first term,

$$\begin{aligned}\mathsf{E}\big[(\widetilde{U}-\widetilde{T})I_{\{\widetilde{T}\geqslant T\}}\big] &= \mathsf{E}\big[(U_{\widetilde{K}}-t_{\widetilde{K}})I_{\{\widetilde{T}\geqslant T\}}\big]\\ &= \sum_{j\geq 0,b\geq 0,x\geq 0}\mathsf{P}(T=j,\widetilde{K}=b,D_b=x)x\cdot I_{\{\frac{b}{r}\geqslant j\}}\\ &= \sum_{j\geq 0,b\geq 0,x\geq 0}\mathsf{P}(T=j,\widetilde{K}=b)\mathsf{P}(D_b=x)x\cdot I_{\{\frac{b}{r}\geqslant j\}},\end{aligned}$$

where we have used the facts that (i) the decision process is based on only what the packets carry and not on their arrival time, and (ii) the condition that sampling is done periodically at a known rate r. Assuming the queueing system to be stationary, the preceding can be written as

$$\begin{aligned}\mathsf{E}\Big[(\widetilde{U}-\widetilde{T})I_{\{\widetilde{T}\geqslant T\}}\Big] &= \left(\sum_{x\geq 0}\mathsf{P}(D=x)x\right)\sum_{j,b}\mathsf{P}(T=j,\widetilde{K}=b)I_{\{\frac{b}{r}\geqslant j\}}\\ &= \mathsf{E}\big[D\big]\mathsf{P}(\widetilde{T}\geqslant T).\end{aligned}$$

Note that $\mathsf{E}[D]$ is a function of the sampling rate r and does not depend on the detection policy.

Consider the second term of Equation (25),

$$\begin{aligned}\mathsf{E}\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{T}\geq T\}}\right] &= \mathsf{E}\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{K}\geq K\}}\right]\\ &= \mathsf{E}\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{K}\geq K,S_0=1\}}\right]+\mathsf{E}\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{K}\geq K,S_0=0\}}\right].\end{aligned}$$

For $S_0=1$, we have $T=0$ and $K=0$. Hence,

$$\mathsf{E}\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{T}\geq T\}}\right] = 0+\mathsf{E}_0\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{K}\geq K\}}\right]$$

where $\mathsf{E}_0[\cdot]$ denote the expectation and $\mathsf{P}_0(\cdot)$ the probability law when the initial state is $S_0=0$. Now,

$$\begin{aligned}\mathsf{E}_0\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{K}\geq K\}}\right] &= \sum_{b=1}^{\infty}\sum_{\widetilde{b}=b}^{\infty}\sum_{t=(b-1)/r+1}^{b/r}\mathsf{P}_0(T=t,K=b,\widetilde{K}=\widetilde{b})\cdot\left(\frac{b}{r}-t\right)\\ &= \sum_{b=1}^{\infty}\sum_{\widetilde{b}=b}^{\infty}\mathsf{P}_0(K=b,\widetilde{K}=\widetilde{b})\\ &\quad\cdot\left[\sum_{t=(b-1)/r+1}^{b/r}\mathsf{P}_0(T=t\mid K=b,\widetilde{K}=\widetilde{b})\cdot\left(\frac{b}{r}-t\right)\right].\end{aligned}\tag{26}$$

We note that $\widetilde{K}$ is independent of T, given K. Hence,

$$\begin{aligned}\mathsf{E}_0\left[\left(\frac{K}{r}-T\right)I_{\{\widetilde{K}\geq K\}}\right] &= \sum_{b=1}^{\infty}\sum_{\widetilde{b}=b}^{\infty}\mathsf{P}_0(K=b,\widetilde{K}=\widetilde{b})\\ &\quad\cdot\left[\sum_{y=0}^{1/r-1}y\cdot\mathsf{P}_0\left(T=\frac{b}{r}-y\mid K=b\right)\right].\end{aligned}$$

We have

$$\mathsf{P}_0\left(T = t \mid K = b\right) = \begin{cases} \frac{(1-\rho)(1-p)^{t-1}p}{(1-\rho)(1-p_r)^{b-1}p_r}, & \text{for } t \text{ s.t. } b = \lceil t \cdot r \rceil \\ 0, & \text{otherwise.} \end{cases}$$

Hence, for $0 \leq y \leq 1/r - 1$,

$$\mathsf{P}_0\left(T = \frac{b}{r} - y \mid K = b\right) = \frac{(1-p)^{b/r-y-1}p}{(1-p_r)^{b-1}p_r},$$

but $(1 - p_r) = (1 - p)^{1/r}$, hence,

$$\begin{aligned}\mathsf{P}_0\left(T = \frac{b}{r} - y \mid K = b\right) &= \frac{(1-p)^{b/r-y-1}p}{(1-p_r)^{b-1}p_r} \\ &= \frac{(1-p)^{1/r-y-1}p}{1-(1-p)^{1/r}}.\end{aligned}$$

It can be shown that

$$\begin{aligned}\sum_{y=0}^{1/r-1} y \cdot \frac{(1-p)^{1/r-y-1}p}{1-(1-p)^{1/r}} &= \frac{1}{r} - \left(\frac{1}{p} - \frac{1}{rp_r}(1-p_r)\right) \\ &=: l(r).\end{aligned}$$

Therefore, Equation (26) can be written as

$$\begin{aligned}\mathsf{E}_0\left[\left(\frac{K}{r} - T\right) I_{\{\widetilde{K} \geq K\}}\right] &= l(r) \cdot \mathsf{P}_0(\widetilde{K} \geq K) = l(r) \cdot (\mathsf{P}(\widetilde{K} \geq K) - \rho) \\ &= l(r) \cdot (1 - \mathsf{P}(\widetilde{K} < K) - \rho).\end{aligned}$$

Finally, we have

$$\begin{aligned}&\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T) I_{\{\widetilde{T} \geq T\}}\right] \\ &= \min_{\Pi_\alpha}\{d(r)(1 - \mathsf{P}(\widetilde{T} < T)) + l(r)\mathsf{P}_0(\widetilde{T} \geq T) + \frac{1}{r}\mathsf{E}\left[(\widetilde{K} - K)^+\right]\} \\ &= \min_{\Pi_\alpha}\left\{\left(d(r) + l(r)\right)\left(1 - \mathsf{P}\left(\widetilde{T} < T\right)\right) - \rho \cdot l(r) + \frac{1}{r}\mathsf{E}\left[(\widetilde{K} - K)^+\right]\right\}.\end{aligned}$$

Note that in the preceding equation, the first term $\left(d(r) + l(r)\right)(1 - \mathsf{P}(\widetilde{T} < T))$ is minimum when $\mathsf{P}(\widetilde{T} < T) = \alpha$. It follows that

$$\begin{aligned}&\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T) I_{\{\widetilde{T} \geq T\}}\right] \\ &\geqslant \left(d(r) + l(r)\right)(1 - \alpha) - \rho \cdot l(r) + \frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{K} - K)^+\right].\end{aligned}$$

Also, since the optimal policy for the problem $\min_{\Pi_\alpha} \mathsf{E}[(\widetilde{K} - K)^+]$ achieves $(1 - \mathsf{P}(\widetilde{T} < T)) = \alpha$, we also have

$$(d(r) + l(r))(1 - \alpha) - \rho \cdot l(r) + \frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{K} - K)^+\right] \geqslant \min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T) I_{\{\widetilde{T} \geq T\}}\right].$$

It follows that

$$\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T) I_{\{\widetilde{T} \geq T\}}\right] = \left(d(r) + l(r)\right)(1 - \alpha) - \rho \cdot l(r) + \frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{K} - K)^+\right].$$

We need $1-\alpha > \rho$ or $\alpha < 1-\rho$. If $\alpha > 1-\rho$, the optimal stopping is at $t=0$. This will yield the desired probability of false alarm and $\mathsf{E}[(\widetilde{U}-T)I_{\{\widetilde{T}\geq T\}}]=0$. □

APPENDIX II

PROOF. (THEOREM 2). The necessity of $Nr < \sigma$ is clear. The sufficiency proof goes as follows. Consider the FJQ-GPS system with every queue always containing a single dummy packet that is served at low priority. Let us call this the saturated FJQ-GPS system. When a queue becomes empty, the low-priority dummy packet contends for service. If it receives service, then it immediately reappears and continues to contend for service. If, while a dummy packet is in service, a regular packet arrives, then the service of the dummy packet is preempted, and the regular packet starts contending. It follows that the service rate applied to every queue (i.e., those with regular packets or those with dummy packets) is always σ/N. Now, consider a virtual service process of rate σ. In each slot, a service occurs with probability σ, and the service is applied to any one of the queues with equal probability. Equivalently, each queue is served by an independent Bernoulli process of rate σ/N. Considering only the services to the regular packets at each queue, we have a $GI/M/1$ queue (here GI refers to a general distribution with independent arrivals, M refers to a Markovian service process, and 1 refers to one server). Hence, the system has proper stationary delay, if and only if $r < \sigma/N$. Also, it can be seen that the delays in the described system (with dummy packets when a queue is empty) upperbound those in the original FJQ-GPS system. Hence, the result follows. □

APPENDIX III

Distribution of state noise $\mathbf{N}$.

Let $\mathbf{q} = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]$. Note that $\mathsf{P}\left(M_k = m \middle| \mathbf{Q}_k = \mathbf{q}, \mathbf{\Theta}_k = \boldsymbol{\theta}\right) = \mathsf{P}\left(M_k = m \middle| \mathbf{Q}_k = \mathbf{q}\right)$ and is given by

$$\mathsf{P}\left(M_k = 0 \middle| \mathbf{Q}_k = \mathbf{q}\right) = \begin{cases} 1 & \text{if } \phi_N(\mathbf{q}) = 0 \\ 1-\sigma & \text{if } \phi_N(\mathbf{q}) > 0 \end{cases},$$

$$\mathsf{P}\left(M_k = m \middle| \mathbf{Q}_k = \mathbf{q}\right) = \begin{cases} 0 & \text{if } \phi_N(\mathbf{q}) = 0 \\ \frac{\sigma}{\phi_N(\mathbf{q})} & \text{if } \phi_{L^{(m)}}(\mathbf{q}) > 0, \quad m = 1, 2, 3, \ldots, N, \end{cases}$$

where $\phi_N(\mathbf{q})$ and $\phi_{L^{(m)}}(\mathbf{q})$ are obtained from Equations (9) and (8).

The distribution function $\mathsf{P}(O_k = o|\mathbf{Q}_k = \mathbf{q}, \mathbf{\Theta}_k = \boldsymbol{\theta}) = \mathsf{P}(O_k = o|\mathbf{Q}_k = \mathbf{q}, \Theta_k = \theta)$ is given by

$$\mathsf{P}\left(O_k = o \middle| \mathbf{Q}_k = \mathbf{q}, \Theta_k = 0\right) = \begin{cases} 1-p & \text{if } o = 0 \\ p & \text{if } o = 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathsf{P}\left(O_k = o \middle| \mathbf{Q}_k = \mathbf{q}, \Theta_k = 1\right) = \begin{cases} 1 & \text{if } o = 0 \\ 0 & \text{otherwise.} \end{cases}$$

APPENDIX IV

PROOF (LEMMA 1).

Let $\mathbf{q} = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]$. From Equation (17),

$$\begin{aligned} \Pi_k &:= \mathsf{P}\left(T \leqslant k \middle| \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\ &= \mathsf{P}\left(T \leqslant k-\delta \middle| \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) + \mathsf{P}\left(k-\delta < T \leqslant k \middle| \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \end{aligned}$$

$$
\begin{aligned}
&= \mathsf{P}\left(T \leqslant k-\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) + \mathsf{P}\left(T > k-\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right) \\
&\quad \cdot \mathsf{P}\left(T \leqslant k \,\middle|\, T > k-\delta, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right), \\
&= \Psi_k + (1-\Psi_k) \cdot \mathsf{P}\left(T \leqslant k \,\middle|\, T > k-\delta, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right), \\
&= \Psi_k + (1-\Psi_k) \cdot \frac{\mathsf{P}(k-\delta < T \leqslant k)\, \mathsf{P}\left(\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, k-\delta < T \leqslant k\right)}{\mathsf{P}(T > k-\delta)\, \mathsf{P}\left(\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, T > k-\delta\right)} \\
&= \Psi_k + (1-\Psi_k) \cdot \frac{\mathsf{P}(k-\delta < T \leqslant k)}{\mathsf{P}(T > k-\delta)} \qquad (27) \\
&= \Psi_k + (1-\Psi_k)\left(1-(1-p)^{\delta}\right). \qquad (28)
\end{aligned}
$$

Equation (27) is justified as follows. Note that

$$
\begin{aligned}
&\mathsf{P}\left(\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, k-\delta < T \leqslant k\right) \\
&= \mathsf{P}\left(\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\}, u_{[0:k-1]} \,\middle|\, k-\delta < T \leqslant k\right) \\
&= \mathsf{P}\left(\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q} \,\middle|\, k-\delta < T \leqslant k\right) \\
&\quad \cdot \mathsf{P}\left(\mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\} \,\middle|\, k-\delta < T \leqslant k, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right) \\
&\quad \cdot \mathsf{P}\left(u_{[0:k-1]} \,\middle|\, k-\delta < T \leqslant k, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\}\right) \\
&= \mathsf{P}\left(\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right) \cdot \mathsf{P}\left(\mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\} \,\middle|\, k-\delta < T, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right) \\
&\quad \cdot \mathsf{P}\left(u_{[0:k-1]} \,\middle|\, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\}\right) \\
&= \mathsf{P}\left(\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q} \,\middle|\, T > k-\delta\right) \\
&\quad \cdot \mathsf{P}\left(\mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\} \,\middle|\, T > k-\delta, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right) \\
&\quad \cdot \mathsf{P}\left(u_{[0:k-1]} \,\middle|\, \middle|\, T > k-\delta, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X^{(i)}_{B_k} : R^{(i)}_k = 1\}\right) \\
&= \mathsf{P}\left(\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, T > k-\delta\right).
\end{aligned}
$$

We use the following facts in the preceding justification: i) the evolution of the queueing system $\mathbf{Q}_k$ is independent of the change point T, (ii) whenever $T > k-\delta$, the distribution of any sample $X^{(i)}_h$, $h \leqslant B_k$ is f_0; and iii) the control $u_k = \tilde{\mu}(\mathbf{I}_k)$. □

APPENDIX V

Recursive computation of Π_k

At time k, based on the index of the node that successfully transmits a packet M_k, the set of all sample paths Ω can be partitioned based on the following events.

$$
\begin{aligned}
\mathcal{E}_{1,k} &:= \left\{\omega : M_k(\omega) = 0 \text{ or } M_k(\omega) = j > 0, R^{(j)}_k(\omega) = 1\right\}, \\
\mathcal{E}_{2,k} &:= \left\{\omega : M_k(\omega) = j > 0, R^{(j)}_k(\omega) = 0, \sum_{i=1}^{N} R^{(i)}_k(\omega) < N-1\right\},
\end{aligned}
$$

$$\mathcal{E}_{3,k} := \left\{\omega : M_k(\omega) = j > 0, R_k^{(j)}(\omega) = 0, \sum_{i=1}^{N} R_k^{(i)}(\omega) = N-1\right\},$$

that is, $\Omega = \mathcal{E}_{1,k} \cup \mathcal{E}_{2,k} \cup \mathcal{E}_{3,k}$. We note that the preceding events can also be described by using $\mathbf{Q}_k$ and $\mathbf{Q}_{k+1}$ in the following manner.

$$\begin{aligned}
\mathcal{E}_{1,k} &= \left\{\omega : \mathbf{W}_{k+1}(\omega) = \mathbf{W}_k(\omega), \mathbf{R}_{k+1}(\omega) = \mathbf{R}_k(\omega)\right\} \\
&\quad \bigcup \left\{\omega : \mathbf{W}_{k+1}(\omega) = \mathbf{W}_k(\omega) + \mathbf{e}_j, \mathbf{R}_{k+1}(\omega) = \mathbf{R}_k(\omega)\right\}, \\
\mathcal{E}_{2,k} &= \left\{\omega : \mathbf{W}_{k+1}(\omega) = \mathbf{W}_k(\omega), \mathbf{R}_{k+1}(\omega) = \mathbf{R}_k(\omega) + \mathbf{e}_j\right\}, \\
\mathcal{E}_{3,k} &= \left\{\omega : \sum_{i=1}^{N} R_k^{(i)}(\omega) = N-1, \forall i, W_{k+1}^{(i)}(\omega) = (W_k^{(i)}(\omega) - 1)^+, R_{k+1}^{(i)}(\omega) = \mathbf{1}_{\{W_k^{(i)}>0\}}\right\}.
\end{aligned}$$

Here, events $\mathcal{E}_{1,k}$ and $\mathcal{E}_{2,k}$ represent case $B_{k+1} = B_k$, and event $\mathcal{E}_{3,k}$ represents case $B_{k+1} = B_k + 1$ (i.e., only if event $\mathcal{E}_{3,k}$ occurs, then the batch index is incremented). We are interested in obtaining Π_{k+1} from $[\mathbf{Q}_k, \Pi_k]$ and $\mathbf{Z}_{k+1}$. We show that at time $k+1$, the statistic Ψ_{k+1} (after having observed $\mathbf{Z}_{k+1}$) can be computed in a recursive manner using Ψ_k and $\mathbf{Q}_k$. Using Lemma 5.1 (using Equation (19)), we compute Π_{k+1} from Ψ_{k+1}.

$$\begin{aligned}
\Psi_{k+1} &= \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \mathbf{I}_{k+1}\right) \\
&= \sum_{c=1}^{3} \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1, \mathcal{E}_{c,k} \mid \mathbf{I}_{k+1}\right) \\
&= \sum_{c=1}^{3} \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{c,k}, \mathbf{I}_{k+1}\right) \mathbf{1}_{\mathcal{E}_{c,k}} \qquad (\because \mathcal{E}_{c,k} \text{ is } \mathbf{I}_{k+1} \text{ measurable}).
\end{aligned}$$

Case $M_k = 0$ or $M_k = j > 0, R_k^{(j)} = 1$.

$$\begin{aligned}
&\Pi_{k+1} \\
&= \mathsf{P}\left(\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_{k+1}\right) \\
&= \mathsf{P}\left(\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_k, \mathbf{Q}_{k+1} = \mathbf{q}'\right) \\
&= \frac{\mathsf{P}\left(\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_k\right) \cdot f_{\mathbf{Q}_{k+1}|\Theta_{k+1},\mathcal{E}_{1,k},\mathbf{I}_k}(\mathbf{q}'|1, \mathcal{E}_{1,k}, \mathbf{I}_k)}{f_{\mathbf{Q}_{k+1}\,\big|\,\mathcal{E}_{1,k},\mathbf{I}_k}(\mathbf{q}'|\mathcal{E}_{1,k}, \mathbf{I}_k)} \qquad \text{(by Bayes rule)} \\
&= \mathsf{P}\left(\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_k\right) \qquad (\mathbf{Q}_{k+1} \text{ is independent of } \Theta_{k+1}) \\
&= \mathsf{P}\left(\Theta_k = 0, \Theta_{k+1} = 1 \mid \mathbf{I}_k\right) + \mathsf{P}\left(\Theta_k = 1, \Theta_{k+1} = 1 \mid \mathbf{I}_k\right) \\
&= (1 - \Pi_k)p + \Pi_k.
\end{aligned}$$

Case $M_k = j > 0$, $R_k^{(j)} = 0$, $\sum_{i=1}^{N} R_k^{(i)} < N-1$. In this case, the sample $X_{B_k}^{(j)}$ is successfully transmitted and is passed on to the decision maker. The decision maker receives just this sample and computes Π_{k+1}. We compute Ψ_{k+1} from Ψ_k and then use Lemma 5.1 using Equation (19)) to compute Π_{k+1} from Ψ_{k+1}.

$$\begin{aligned}
&\Psi_{k+1} \\
&= \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_{k+1}\right) \\
&= \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right) \\
&= \mathsf{P}\left(\widetilde{\Theta}_k = 0, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right) \\
&\quad + \mathsf{P}\left(\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right).
\end{aligned}$$

Since we consider the case in which the fusion center receives a sample at time $k+1$ and $B_{k+1} = B_k$, $\Delta_{k+1} = \Delta_k + 1$, and hence, the state $\widetilde{\Theta}_{k+1} = \Theta_{k+1-\Delta_{k+1}} = \Theta_{k-\Delta_k} = \widetilde{\Theta}_k$. Thus, in this case, Ψ_{k+1} can be written as

$$\begin{aligned}
&\Psi_{k+1}\\
&= \mathsf{P}\left(\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right)\\
&\stackrel{(a)}{=} \frac{\mathsf{P}\left(\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k\right) \cdot \mathsf{P}\left(\mathbf{Q}_{k+1} = \mathbf{q}' \mid \widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1, \mathcal{E}_{2,k}, \mathbf{I}_k\right)}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{2,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{2,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y|\mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{q}')}\\
&\quad \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k,\widetilde{\Theta}_{k+1},\mathcal{E}_{2,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y \mid 1, 1, \mathcal{E}_{2,k}, \mathbf{q}', \mathbf{I}_k)\\
&\stackrel{(b)}{=} \frac{\mathsf{P}\left(\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k\right) \cdot \mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{2,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k}(y \mid 1)}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{2,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{2,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y|\mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{q}')}\\
&\stackrel{(c)}{=} \frac{\mathsf{P}\left(\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k\right) \cdot f_1(y)}{\mathsf{P}\left(\widetilde{\Theta}_k = 0 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}\right) \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k}(y|0) + \mathsf{P}\left(\widetilde{\Theta}_k = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}\right) \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k}(y|1)}\\
&\stackrel{(d)}{=} \frac{\Psi_k f_1(y)}{(1-\Psi_k) f_0(y) + \Psi_k f_1(y)}.
\end{aligned}$$

We explain the steps (a), (b), (c), and (d) below.

(a) By Bayes rule, for events A, B, C, D, E, and F, we have

$$\mathsf{P}(AB \mid CDEF) = \frac{\mathsf{P}(AB \mid CD)\,\mathsf{P}(E \mid ABCD)\,\mathsf{P}(F \mid ABCDE)}{\mathsf{P}(E \mid CD)\,\mathsf{P}(F \mid CDE)}.$$

(b) $\mathbf{Q}_{k+1}$ is independent of $\widetilde{\Theta}_k$, $\widetilde{\Theta}_{k+1}$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1}$ is independent of $\widetilde{\Theta}_{k+1}, \mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$.

(c) For any events A, B, and a continuous random variable Y, the conditional density function $f_{Y|A}(y|A) = \mathsf{P}(B \mid A)\, f_{Y|AB}(y|AB) + \mathsf{P}(B^c \mid A)\, f_{Y|AB^c}(y|AB^c)$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1}$ is independent of $\mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$.

(d) $\mathcal{E}_{2,k}$ is $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$ measurable, and hence, given $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$, $\widetilde{\Theta}_k$ is independent of $\mathcal{E}_{2,k}$.

Case $M_k = j > 0$, $R_k^{(j)} = 0$, $\sum_{i=1}^{N} R_k^{(i)} = N-1$. In this case, at time $k+1$, the decision maker receives the last sample of batch B_k, $X_{B_k}^{(j)}$ (that is successfully transmitted during slot k) and the samples of batch B_k+1, if any, that are queued in the sequencer buffer. We compute Ψ_{k+1} from Ψ_k and use Lemma 1 (using Equation (19)) to compute Π_{k+1} from Ψ_{k+1}. In this case, the decision maker receives $n := \sum_{i=1}^{N} \mathbf{1}_{\{W_k^{(i)} > 0\}}$ samples of batch B_k+1. Also, note that n is $\mathbf{I}_k$ measurable.

$$\begin{aligned}
\Psi_{k+1} &= \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_{k+1}\right)\\
&= \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right)\\
&= \mathsf{P}\left(\widetilde{\Theta}_k = 0, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right)\\
&\quad + \mathsf{P}\left(\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right).
\end{aligned}$$

Since, we consider the case $B_{k+1} = B_k + 1$, $\Delta_{k+1} = \Delta_k + 1 - 1/r$, and hence, the state $\widetilde{\Theta}_{k+1} = \Theta_{k+1-\Delta_{k+1}} = \Theta_{k-\Delta_k+1/r}$.

Let $\mathbf{y} = [y_0, y_1, \dots, y_n]$. Consider

$$\begin{aligned}
&\mathsf{P}\left(\widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right)\\
&\stackrel{(a)}{=} \frac{\mathsf{P}\left(\widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k\right) \cdot \mathsf{P}\left(\mathbf{Q}_{k+1} = \mathbf{q}' \mid \widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1, \mathcal{E}_{3,k}, \mathbf{I}_k\right)}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{3,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{3,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(\mathbf{y} | \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{q}')}\\
&\quad \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k,\widetilde{\Theta}_{k+1},\mathcal{E}_{3,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(\mathbf{y} \mid \widetilde{\theta}, 1, \mathcal{E}_{3,k}, \mathbf{q}', \mathbf{I}_k)\\
&\stackrel{(b)}{=} \frac{\mathsf{P}\left(\widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k\right) \cdot \mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{3,k}, \mathbf{I}_k) \cdot f_{\widetilde{\theta}}(y_0) \prod_{i=1}^n f_1(y_i)}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{3,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{3,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(\mathbf{y} | \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{q}')}\\
&\stackrel{(c)}{=} \frac{\mathsf{P}\left(\widetilde{\Theta}_k = \widetilde{\theta} \mid \mathcal{E}_{3,k}, \mathbf{I}_k\right) \cdot \mathsf{P}\left(\widetilde{\Theta}_{k+1} = 1 \mid \widetilde{\Theta}_k = \widetilde{\theta}, \mathcal{E}_{3,k}, \mathbf{I}_k\right) \cdot f_{\widetilde{\theta}}(y_0) \prod_{i=1}^n f_1(y_i)}{\sum_{\widetilde{\theta}'=0}^1 \sum_{\widetilde{\theta}''=0}^1 \mathsf{P}\left(\widetilde{\Theta}_k = \widetilde{\theta}', \widetilde{\Theta}_{k+1} = \widetilde{\theta}'', \mid \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}\right) \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k,\widetilde{\Theta}_{k+1}\mathcal{E}_{3,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y | \widetilde{\theta}', \widetilde{\theta}'', \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{q}')}.
\end{aligned}$$

We explain the steps (a), (b), and (c) below.

(a) By Bayes rule, for events A, B, C, D, E, and F, we have

$$\mathsf{P}(AB \mid CDEF) = \frac{\mathsf{P}(AB \mid CD)\, \mathsf{P}(E \mid ABCD)\, \mathsf{P}(F \mid ABCDE)}{\mathsf{P}(E \mid CD)\, \mathsf{P}(F \mid CDE)}.$$

(b) $\mathbf{Q}_{k+1}$ is independent of $\widetilde{\Theta}_k$, $\widetilde{\Theta}_{k+1}$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1,0}$ is independent of $\widetilde{\Theta}_{k+1}, \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$, and given $\widetilde{\Theta}_{k+1}$, $\mathbf{Y}_{k+1,i}$ is independent of $\widetilde{\Theta}_k, \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$. It is to be noted that given the state of nature, the sensor measurements $Y_{k+1,0}, Y_{k+1,1}, \dots, Y_{k+1,n}$ are conditionally independent.

(c) For any events A, B, and a continuous random variable Y, the conditional density function $f_{Y|A}(y|A) = \mathsf{P}(B \mid A)\, f_{Y|AB}(y|AB) + \mathsf{P}(B^c \mid A)\, f_{Y|AB^c}(y|AB^c)$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1}$ is independent of $\mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$.

It is to be noted that the event $\mathcal{E}_{3,k}$ is $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$ measurable, and hence, given $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$, $\widetilde{\Theta}_k$ is independent of $\mathcal{E}_{3,k}$. Thus, in this case,

$$\Psi_{k+1} = \frac{(1-\Psi_k)p_r f_0(y_0) \prod_{i=1}^n f_1(y_i) + \Psi_k f_1(y_0) \prod_{i=1}^n f_1(y_i)}{(1-\Psi_k)(1-p_r) f_0(y_0) \prod_{i=1}^n f_0(y_i) + (1-\Psi_k)p_r f_0(y_0) \prod_{i=1}^n f_1(y_i) + \Psi_k f_1(y_0) \prod_{i=1}^n f_1(y_i)}.$$

Thus, using Lemma 5.1 (using Equation (19)), we have

$$\begin{aligned}
\Pi_{k+1} &= \Psi_{k+1} + (1-\Psi_{k+1})(1-(1-p)^{\Delta_{k+1}})\\
&=: \phi_\Psi(\Psi_k, \mathbf{Z}_{k+1}) + \left(1 - \phi_\Psi(\Psi_k, \mathbf{Z}_{k+1})\right)(1-(1-p)^{\Delta_{k+1}})\\
&= \phi_\Psi\left(\frac{\Pi_k - (1-(1-p)^{\Delta_k})}{(1-p)^{\Delta_k}}, \mathbf{Z}_{k+1}\right)\\
&\quad + \left(1 - \phi_\Psi\left(\frac{\Pi_k - (1-(1-p)^{\Delta_k})}{(1-p)^{\Delta_k}}, \mathbf{Z}_{k+1}\right)\right)(1-(1-p)^{\Delta_{k+1}})\\
&=: \phi_\Pi\left([\mathbf{Q}_k, \Pi_k], \mathbf{Z}_{k+1}\right).
\end{aligned}$$

APPENDIX VI

Structure of τ^*.

We use the following Lemma to show that $J^*(\mathbf{q}, \pi)$ is concave in π.

LEMMA 2. *If $f : [0,1] \to \mathbb{R}$ is concave, then the function $h : [0,1] \to \mathbb{R}$, defined by*

$$h(y) = \mathsf{E}_{\phi(\mathbf{x})}\left[f\left(\frac{y \cdot \phi_2(\mathbf{x}) + (1-y)p_r \cdot \phi_1(\mathbf{x})}{y \cdot \phi_2(\mathbf{x}) + (1-y)p_r \cdot \phi_1(\mathbf{x}) + (1-y)(1-p_r) \cdot \phi_0(\mathbf{x})}\right)\right],$$

is concave for each $\mathbf{x}$, where $\phi(\mathbf{x}) = y \cdot \phi_2(\mathbf{x}) + (1-y)p_r \cdot \phi_1(\mathbf{x}) + (1-y)(1-p_r) \cdot \phi_0(\mathbf{x})$, $0 < p_r < 1$, and $\phi_0(\mathbf{x})$, $\phi_1(\mathbf{x})$, and $\phi_2(\mathbf{x})$ are pdfs on $\mathbf{X}$.

PROOF. See Appendix I of Premkumar and Kumar [2008].

Note that in the finite H-horizon (truncated version of Equation (21)), we note from value iteration that the cost-to-go function for a given $\mathbf{q}$ $J_H^H([\mathbf{q}, \pi]) = 1-\pi$ is concave in π. Hence, by Lemma 2, we see that for any given $\mathbf{q}$, the cost-to-go functions $J_{H-1}^H([\mathbf{q}, \pi])$, $J_{H-2}^H([\mathbf{q}, \pi])$, $\ldots$, $J_0^H([\mathbf{q}, \pi])$ are concave in π. Hence for $0 \le \lambda \le 1$,

$$\begin{aligned}
J^*([\mathbf{q}, \pi]) &= \lim_{H\to\infty} J_0^H([\mathbf{q}, \pi]) \\
J^*([\mathbf{q}, \lambda\pi_1 + (1-\lambda)\pi_2]) &= \lim_{H\to\infty} J_0^H\Big([\mathbf{q}, \lambda\pi_1 + (1-\lambda)\pi_2]\Big) \\
&\ge \lim_{H\to\infty} \lambda J_0^H([\mathbf{q}, \pi_1]) + \lim_{H\to\infty} (1-\lambda) J_0^H([\mathbf{q}, \pi_2]) \\
&= \lambda J^*([\mathbf{q}, \pi_1]) + (1-\lambda) J^*([\mathbf{q}, \pi_2]).
\end{aligned}$$

It follows that for any given $\mathbf{q}$, $J^*([\mathbf{q}, \pi])$ is concave in π. □

Define the map $\xi : \mathcal{Q}\times[0,1] \to \mathbb{R}_+$ as $\xi([\mathbf{q}, \pi]) := 1-\pi$ and the map $\kappa : \mathcal{Q}\times[0,1] \to \mathbb{R}_+$, as $\kappa([\mathbf{q}, \pi]) := c\cdot\pi + A_{J^*}([\mathbf{q}, \pi]) = c\cdot\pi + \mathsf{E}[J^*([\mathbf{Q}_{k+1}, \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})])\,|\,\nu_k = [\mathbf{q}, \pi]]$. Note that $\xi([\mathbf{q}, 1]) = 0$, $\kappa([\mathbf{q}, 1]) = c$, $\xi([\mathbf{q}, 0]) = 1$ and

$$\begin{aligned}
&\kappa([\mathbf{q}, 0]) \\
&\quad= \mathsf{E}[J^*([\mathbf{Q}_{k+1}, \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})])|\nu_k = [\mathbf{q}, 0]] \\
&\quad\overset{(2)}{=} \mathsf{E}[J^*([\phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k), \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})])|\nu_k = [\mathbf{q}, 0]] \\
&\quad= \sum_{m=0}^{N} \mathsf{E}[J^*([\phi_{\mathbf{Q}}(\mathbf{q}, m), \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})])|M_k = m, \nu_k = [\mathbf{q}, 0]]\mathsf{P}(M_k = m|\nu_k = [\mathbf{q}, 0]) \\
&\quad\overset{(4)}{\leqslant} \sum_{m=0}^{N} J^*([\phi_{\mathbf{Q}}(\mathbf{q}, m), \mathsf{E}[\phi_\Pi(\nu_k, \mathbf{Z}_{k+1})|M_k = m, \nu_k = [\mathbf{q}, 0]]])\mathsf{P}(M_k = m|\nu_k = [\mathbf{q}, 0]) \\
&\quad= \sum_{m=0}^{N} J^*([\phi_{\mathbf{Q}}(\mathbf{q}, m), p)\mathsf{P}\left(M_k = m\,\middle|\,\nu_k = [\mathbf{q}, 0]\right) \\
&\quad\overset{(6)}{\leqslant} \sum_{m=0}^{N} (1-p)\cdot\mathsf{P}\left(M_k = m\,\middle|\,\nu_k = [\mathbf{q}, 0]\right) \\
&\quad= 1-p \;<\; 1,
\end{aligned}$$

where in the preceding derivation, we use the evolution of $\mathbf{Q}_k$ in step 2, the Jensen's inequality (as for any given $\mathbf{q}$, $J^*(\mathbf{q}, \pi)$ is concave in π) in step 4, and the inequality $J^*(\mathbf{q}, \pi) \leqslant 1-\pi$ in step 6.

Note that $\kappa([\mathbf{q}, 1]) - \xi([\mathbf{q}, 1]) > 0$ and $\kappa([\mathbf{q}, 0]) - \xi([\mathbf{q}, 0]) < 0$. Also, for a fixed $\mathbf{q}$, the function $\kappa([\mathbf{q}, \pi]) - \xi([\mathbf{q}, \pi])$ is concave in π. Hence, by the *intermediate value theorem*, for a fixed $\mathbf{q}$, there exists $\gamma(\mathbf{q}) \in [0,1]$ such that $\kappa([\mathbf{q}, \gamma]) = \xi([\mathbf{q}, \gamma])$. This γ is unique as $\kappa([\mathbf{q}, \pi]) = \xi([\mathbf{q}, \pi])$ for at most two values of π. If in the interval $[0,1]$ there are two distinct values of π for which $\kappa([\mathbf{q}, \pi]) = \xi([\mathbf{q}, \pi])$, then the signs of $\kappa([\mathbf{q}, 0]) - \xi([\mathbf{q}, 0])$ and $\kappa([\mathbf{q}, 1]) - \xi([\mathbf{q}, 1])$ should be the same. Hence,

$$\tau^* = \inf\left\{k : \Pi_k \geqslant \gamma(\mathbf{Q}_k)\right\},$$

where the threshold $\gamma(\mathbf{q})$ is given by $c\cdot\gamma(\mathbf{q}) + A_{J^*}([\mathbf{q}, \gamma(\mathbf{q})]) = 1-\gamma(\mathbf{q})$. □

REFERENCES

ALDOSARI, S. A. AND MOURA, J. M. F. 2004. Detection in decentralized sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 277–280.

BACCELLI, F. AND MAKOWSKI, A. 1990. Synchronization in queueing systems. In *Stochastic Analysis of Computer and Communication Systems*, H. Takagi, Ed. North–Holland, 57–131.

BERTSEKAS, D. P. 2000a. *Dynamic Programming and Optimal Control*, 2nd Ed. vol. I. Athena Scientific.

BERTSEKAS, D. P. 2000b. *Dynamic Programming and Optimal Control*, Second ed. Vol. II. Athena Scientific.

HONEYWELL INC. http://hpsweb.honeywell.com/Cultures/en-US/Products/Wireless/SecondGeneration-Wireless/default.htm.

ISA. http://www.isa.org/Content/NavigationMenu/Technical_Information/ASCI/ISA100_Wireless_Compliance_Institute/ISA100_Wireless_Compliance_Institute.htm.

KUMAR, A., MANJUNATH, D., AND KURI, J. 2004. *Communication Networking: An Analytical Approach*. Morgan-Kaufmann, San Francisco.

KUMAR, A., MANJUNATH, D., AND KURI, J. 2008. *Wireless Networking*. Morgan-Kaufmann, San Francisco.

NIU, R. AND VARSHNEY, P. K. 2005. Distributed detection and fusion in a large wireless sensor network of random size. *EURASIP J. Wirel. Comm. Netw. 4,* 7, 462–472.

PREMKUMAR, K. AND KUMAR, A. 2008. Optimal sleep-wake scheduling for quickest intrusion detection using wireless sensor networks. In *Proceedings of the International Conference on Computer Communications IEEE (INFOCOM)*.

PREMKUMAR, K., KUMAR, A., AND KURI, J. 2009. Distributed detection and localization of events in large ad hoc wireless sensor networks. In *Proceedings of the* 47^{th} *Annual Allerton Conference on Communication, Control, and Computing*.

RAJASEGARAR, S., LECKIE, C., AND PALANISWAMI, M. 2008. Anomaly detection in wireless sensor networks. *IEEE Wirel. Comm. 15,* 4, 34–40.

SHIRYAEV, A. N. 1978. *Optimal Stopping Rules*. Springer, New York.

SINGH, C. K., KUMAR, A., AND AMEER, P. M. 2008. Performance evaluation of an IEEE 802.15.4 sensor network with star topology. *Wirel. Netw. 14,* 4, 543–568.

TARTAKOVSKY, A. G. AND VEERAVALLI, V. V. 2005. General asymptotic bayesian theory of quickest change detection. *SIAM Theory Probab. Appl. 49,* 3, 458–497.

TENNY, R. R. AND SANDELL, N. R. 1981. Detection with distributed sensors. *IEEE Trans. Aerosp. Electron. Syst. 17*, 501–510.

VEERAVALLI, V. V. 2001. Decentralized quickest change detection. *IEEE Trans. Inf. Theory 47,* 4, 1657–1665.

Received March 2009; revised December 2009, June, October 2010; accepted October 2010