

Collaborative Quickest Detection in Adhoc Networks with Delay Constraint -Part II:Multi-node Network

Chengzhi Li, Husheng Li and Huaiyu Dai

Abstract—In the first part of this two-part paper, we initiate the study of collaborative quickest detection in an adhoc setting for a simple two-node scenario. In this part, we extend our study to the general multi-node network, where we also explicitly address information exchange protocol and network topology control. We further consider block transmission mode and correlated observations. In each scenario, corresponding multi-thread Cumulative Sum (CUSUM) algorithms and performance analysis are given. Numerical results are provided to verify the theoretical analysis.

I. INTRODUCTION

In the first part of this paper, we have initiated a study on quickest detection in adhoc networks with communication delay, by considering a simple two-node network. In this part, We extend our study to the general multiple node scenario in this paper. We first consider continuous transmission mode as in Part I, where information is exchanged among nodes in every time slot and forms a continuous data stream. We then consider block transmission mode due to the practical requirement of channel coding and its better energy efficiency, where data is transmitted in blocks so collaborative information arrives at a node in batch intermittently. In addition, we also analyze the performance of collaborative quickest detection with correlated observations, which gives some interesting insights.

The remainder of this paper is organized as follows. Section II introduces the system model of a general N -node network and performance measurement. In Section III, continuous transmission mode is considered when we extend our study in Part I to the general case, and block transmission mode is discussed in Section IV. Section V further investigates correlated observations. Numerical results and conclusions are provided in Sections VI and VII, respectively.

Below are some notations used in this paper:

- $[x]^+ = \max(x, 0)$.
- $E_t[\cdot]$ means the expectation conditioned that the change happens at time t . $V_t[\cdot]$ means the variance conditioned that the change happens at time t .
- For a vector \mathbf{x} , $\mathbf{x}(n : m)$ ($n < m$) means the vector containing the n -th to the m -th element of \mathbf{x} .
- $\lfloor x \rfloor$ means the largest integer smaller than or equal to x ;
 $\lceil x \rceil$ means the smallest integer larger than or equal to x .

Chengzhi Li is with the department of electrical and computer engineering, North Carolina State University, Raleigh, NC, (email:lichengzhi@gmail.com)

Husheng Li is with the department of electrical engineering and computer science, the University of Tennessee, Knoxville, TN, (email:husheng@ece.utk.edu)

Huaiyu Dai is with the department of electrical and computer engineering, North Carolina State University, Raleigh, NC, (email:hdai@ncsu.edu)

The work of Chengzhi Li and Huaiyu Dai was supported in part by the National Science Foundation under Grant CCF-0515164 and CNS-0721815

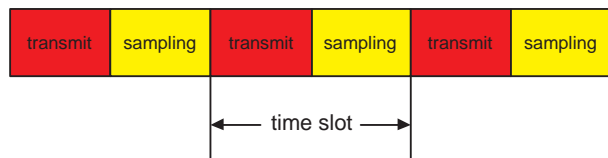


Fig. 1: Structure of time slots

II. SYSTEM MODEL

A. N -Node Network

As in Part I, suppose that an N -node network is monitoring some property of the environment and each node samples the environment using the same frequency. The observation of node θ at the sample period t is given by $X_\theta(t)$. The corresponding probability density function under two possible hypotheses H_0 and H_1 at node θ is denoted by $f_\theta^{(0)}$ and $f_\theta^{(1)}$ respectively. For simplicity, we will mainly assume that $f_\theta^{(0)}$ and $f_\theta^{(1)}$ are both Gaussian. The corresponding log likelihood ratio is defined as

$$l_\theta(t) \triangleq \log \left(\frac{f_\theta^{(1)}(X_\theta(t))}{f_\theta^{(0)}(X_\theta(t))} \right). \quad (1)$$

A graph is used to represent the network, in which a vertex denotes a node and an edge means that the two vertices (nodes) can communicate with each other reliably. If the network topology is not a tree, the likelihood ratio $l(r)$ of a node may be duplicated when being transmitted to other nodes if it is in a cycle of the network. Then, it requires to store all log likelihood ratio values to remove the duplicated version. Therefore, we assume that a spanning tree of the network has been generated and each node multicasts to only its neighboring nodes in the spanning tree. We adopt a time slot structure given in Fig. 1, the detailed description of which can be found in Part I.

We assume that the communication channel is sufficiently good such that all transmit and receive procedures can be completed during the transmit period and there exists a delay of D time slots for each hop (here a hop means an edge in the spanning tree). This forms pipelining flows of log likelihood ratios within the network. We denote the maximum possible delay by D_{\max} , which equals Dd where d is the diameter of the spanning tree.

Below are some notations for the network topology:

- For a node A within the network, denote by $\Theta_A^{(n)}$ the set of nodes being n hops away from node A .
- $N(A)$ denotes the set of the neighboring nodes of node A in the spanning tree and $|N(A)|$ denotes the cardinality of the set $N(A)$.

- $E(A)$ denotes the set of edges incident to node A .

B. Performance Measurement

As in part I, We use two average run lengths (ARL) to measure the performance. For node θ , these ARLs are defined as

$$\mathcal{D}_\theta = \text{esssup} (E_t [T_\theta^* - t | \mathcal{F}_{t-1}]), \quad (2)$$

$$\mathcal{F}_\theta = E_\infty [T_\theta^*], \quad (3)$$

where T_θ^* is the stopping time that node θ determines that a distribution change has happened. \mathcal{F}_{t-1} is the filtration, namely the smallest σ -field with respect to $X_\theta(0), \dots, X_\theta(t-1)$. E_∞ is the expectation under the assumption that the change never happens (the change time is ∞). Obviously, we desire a small \mathcal{D}_θ and a large \mathcal{F}_θ .

III. CONTINUOUS TRANSMISSION

In continuous transmission mode, each sensor sends its likelihood ratio to others every time slot and data arrives at a node continuously. We assume the observations of all sensors are independent both in time and in space. These assumptions accord with our study in Part I.

A. Multi-thread CUSUM test

Similar to the two-thread CUSUM test in part I, we can use the following multi-thread CUSUM test with window size W (window size W means that a node does not use other nodes' observations out of the time window; we suppose $W = wD \leq D_{\max}$ and w is an integer), in which the stopping time for node A is given by

$$T_A^* = \min (T_A^0, \dots, T_A^{w-1}), \quad (4)$$

where, $\forall i = 1, \dots, w-1$,

$$T_A^i = \min \left(\tau \left| m_A^i(\tau - iD) + \sum_{j=0}^{i-1} \sum_{\theta \in \Theta_A^{(j)}} \sum_{r=\tau-iD+1}^{\tau-jD} l_\theta(r) \geq \gamma_A \right. \right), \quad (5)$$

in which

$$m_A^i(t) = \max \left(m_A^i(t-1) + \sum_{j=0}^i \sum_{\theta \in \Theta_A^{(j)}} l_\theta(t), 0 \right), \quad (6)$$

and

$$T_A^0 = \min \left(\tau \left| \max_{\tau-D+1 \leq k \leq \tau} \sum_{r=k}^{\tau} l_\theta(r) \geq \gamma_A \right. \right). \quad (7)$$

It is easy to check that the two-thread algorithm is a special case of the above multi-thread test.

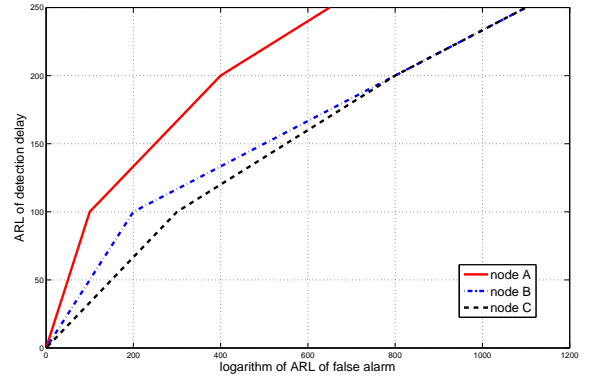


Fig. 2: Performance tradeoff curves of a linear 3-node network

B. Performance Analysis

We assume that a spanning tree has been constructed. Then, by repeating the argument in two-node network, we have the following equation for node A ,

$$I_A \mathcal{D}_A + \sum_{n=1}^w \left(\sum_{\theta \in \Theta_A^{(n)}} I_\theta \right) [\mathcal{D}_A - nD]^+ \doteq \log \mathcal{F}_A, \quad (8)$$

where I_θ is the Kullback-Leibler divergence of node θ defined as

$$I_\theta = E_r [l_\theta(r)]. \quad (9)$$

Figure 2 illustrates the performance of a three-node network, in which nodes A, B and C have Kullback-Leibler divergences 1, 2 and 3, respectively. Suppose that its spanning tree contains edges AB and BC. We observe that the system performance is represented by a family of piecewisely linear curves obtained from (8). Note that, in such a figure, a lower curve implies better performance since the same \mathcal{D} can achieve lower \mathcal{F} .

C. Information Exchange Protocol

Each node maintains a W -dimensional column vector \mathbf{v}_θ , where θ is the label of the node and W is the window size. We call it *new information vector*, which stores the newest information obtained from other nodes. We also need a transmit vector \mathbf{t}_θ , which stores the information to be transmitted, and a memory vector \mathbf{m}_θ , which stores the transmit vector in the previous time slot.

Now, we focus on node A . The repeated procedure of information exchange is given below.

- Before the first time slot, we set $\mathbf{v}_A = 0$ and $\mathbf{m}_A = 0$.
- At time slot t , during the transmit period, node A multicasts the following transmit vector to its neighbors in the spanning tree:

$$\mathbf{t}_A = \begin{pmatrix} l_A \\ \mathbf{v}_A(1 : W-1) \end{pmatrix}, \quad (10)$$

where l_A is the newest log likelihood ratio ready for transmission at node A .

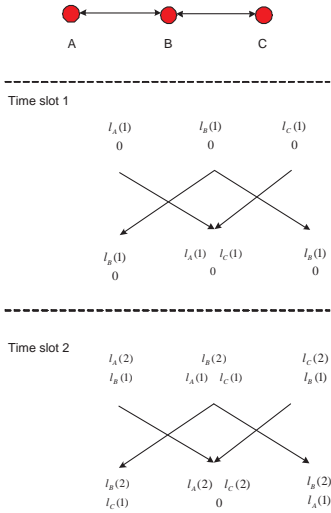


Fig. 3: Illustration of information exchange protocol

- At time slot t , during the sampling period, node A obtains a new log likelihood ratio $l_A(t)$. Meanwhile, node A aggregates the new received information from neighbors and updates the new information vector to

$$\mathbf{v}_A = \sum_{\theta \in N(A)} \mathbf{v}_\theta - |N(A)| \begin{pmatrix} 0 \\ \mathbf{m}_A \\ (1 : W - 1) \end{pmatrix}. \quad (11)$$

Note that the information \mathbf{m}_A stored in the memory vector has been transmitted to all neighbors in time slot $t - 1$ and is transmitted back to node A at time slot t . Therefore, we subtract $|N(A)| (0, \mathbf{m}_A^T(1 : W - 1))^T$ from \mathbf{v}_A to remove the duplication. This removes the duplicated statistics. Finally, we store the transmitted vector at the memory vector, namely $\mathbf{m}_A = \mathbf{t}_A$.

A simple three-node network is illustrated in Fig. 3. The nodes are denoted by A, B and C. We suppose that the first observations are sampled at time slot 1 and the window size is $W = 2$. From the illustrated procedure, we observe that the duplicated statistics are removed and the log likelihood ratio is conveyed from A to C (also from C to A), with the relay of node B in two time slots.

D. Generating Spanning Tree

In previous subsections, we assume that a spanning tree has been generated. However, there are multiple choices of spanning trees and the selection of spanning tree can considerably affect the system performance. We use the following minimax criterion for generating spanning tree, which is given by

$$\mathcal{T}^* = \min_{\mathcal{T}} \max_{\theta} \mathcal{D}_\theta(\mathcal{T}, \tau_0), \quad (12)$$

where \mathcal{T}^* is the optimal spanning tree, \mathcal{T} is a generic spanning tree and $\mathcal{D}_\theta(\mathcal{T}, \tau_0)$ is the average ARL of detection delay of node θ under H_1 , determined by the spanning tree \mathcal{T} and within the constraint

$$\mathcal{F}_\theta \geq \tau_0, \quad \theta \in \Theta,$$

where τ_0 is predetermined constraint. Intuitively, the optimal spanning tree minimizes the maximal ARL of detection delay

under H_1 while assuring that all ARLs of false alarm under H_0 are larger than a threshold.

Notice that, once the spanning tree \mathcal{T} and the threshold τ_0 have been determined, we can compute \mathcal{F}_1 approximately for all nodes using (8). Therefore, we can check all possible spanning trees and select the optimal one. However, this is computational prohibitive for a large network since the number of possible spanning trees is large (for an N -complete graph, the number of spanning trees is N^{N-1}). Therefore, we use the following heuristic incremental algorithm, which tries to minimize the maximal \mathcal{F} in each step:

- Initialization: initialize the edge set of the tree \mathcal{T} as a null set.
- Metric broadcast: each node uses (8) and the current tree topology to compute its \mathcal{F} and then broadcasts it as a metric to all other nodes via a special control channel (suppose that the broadcast can cover all nodes). Note that if a node has no edge to select (adding one more edge results in a cycle), it does not broadcast its metric.
- Edge selection: the node with the smallest metric, denoted by A , gets the priority to select an edge e^* incident to itself, which is given by

$$e^* = \arg \min_{e \in E(A) \cap \mathcal{T}^c} \mathcal{F}_{n(e)}, \quad (13)$$

where \mathcal{T}^c is the complement set of \mathcal{T} and $n(e)$ denotes the node on the other end of edge e . Intuitively, the selection of edges tries to aid the nodes with bad performance. Update \mathcal{T} and repeat the metric broadcast and edge selection until a spanning tree is constructed.

Note that the network for quickest detection is usually fixed. Therefore, the above procedure of spanning tree selection can be carried out when the network is established. The procedure is fast since the number of steps is equal to the number of edges in the spanning tree, $N - 1$.

IV. BLOCK TRANSMISSION

In practice, data are typically collected into blocks and sent in batch for better energy efficiency. Assuming a block length L , i.e., likelihood ratio of L consecutive time slots are transmitted every L time slots. Again, we assume the observations are independent in both time and space.

A. Multi-thread CUSUM Test in Block Transmission Mode

The test procedure in III-A can be readily modified as follows. The stopping time for node A is given by

$$T_A^* = \min (T_A^0, \dots, T_A^{w-1}), \quad (14)$$

where

$$T_A^0 = \min \left(\tau \left| \max_{\lfloor \frac{\tau-D}{L} \rfloor L + 1 \leq k \leq \tau} \sum_{r=k}^{\tau} l_A(r) \geq \gamma_A \right. \right), \quad (15)$$

and $\forall i = 1, \dots, w - 1$,

$$T_A^i = \min \left(\tau \left| m_A^i (\lfloor \frac{\tau - iD}{L} \rfloor L) + n_A^i(\tau) \geq \gamma_A \right. \right), \quad (16)$$

in which

$$n_A^i(\tau) = \sum_{j=1}^{i-1} \sum_{\theta \in \Theta_A^{(j)}} \sum_{r=\lfloor \frac{\tau-jD}{L} \rfloor L+1}^{\lfloor \frac{\tau-jD}{L} \rfloor L} l_\theta(r) + \sum_{r=\lfloor \frac{\tau-iD}{L} \rfloor L+1}^{\tau} l_A(r). \quad (17)$$

In (16) $m_A^i(t)$ is updated every L time slots and in each update it is calculated recursively with n from $L-1$ to 0 as

$$m_A^i(t-n) = \max(m_A^i(t-n-1) + s_A^i(t-n), 0), \quad (18)$$

where

$$s_A^i(t) = \sum_{j=0}^i \sum_{\theta \in \Theta_A^{(j)}} l_\theta(t). \quad (19)$$

B. Performance Analysis

Similar to (8), we have the following equation for sensor A,

$$I_A \mathcal{D}_A + \sum_{n=1}^w \left(\sum_{\theta \in \Theta_A^{(n)}} I_\theta \right) \lfloor \frac{[\mathcal{D}_A - nD]^+}{L} \rfloor L \doteq \log \mathcal{F}_A. \quad (20)$$

Compared with (8), block transmission degrades the performance, but not too much if block length L is of lower order with respect to delay D .

V. CORRELATED OBSERVATIONS

As another extension, we consider the case that observations of different sensors are mutually dependent. In this study, joint distribution of spatially correlated observations are needed. While this may be mainly of theoretical interest, we also concretize our results for the one-dimensional joint Gaussian case and obtain some interesting insights. For simplicity, we only consider continuous transmission; extension to the block transmission scenario is straightforward and omitted. For a node A within the network, denote by $\Phi_A^{(k)}$ the set of nodes from k to 0 hops away from node A , whose observation vector at time slot t is $X_{\Phi_A^{(k)}}(t)$ and joint probability density function is $f_{\Phi_A^{(k)}}^{(i)}$ under hypothesis H_i . The corresponding log likelihood ratio is $l_\Phi(t) \triangleq \log \left(\frac{f_{\Phi_A^{(k)}}^{(1)}(X_\Phi(t))}{f_{\Phi_A^{(k)}}^{(0)}(X_\Phi(t))} \right)$.

A. Multi-Thread CUSUM Test with Correlated Observations

We modify the test procedure in III-A as follows. The stopping time for node A is given by

$$T_A^* = \min(T_A^0, \dots, T_A^{w-1}), \quad (21)$$

where

$$T_A^0 = \min \left(\tau \left| \max_{\tau-D+1 \leq k \leq \tau} \sum_{r=k}^{\tau} l_A(r) \geq \gamma_A \right. \right), \quad (22)$$

and $\forall i = 1, \dots, w-1$,

$$T_A^i = \min \left(\tau \left| m_A^i(\tau - iD) + n_A^i(\tau) \geq \gamma_A \right. \right), \quad (23)$$

in which

$$m_A^i(t) = \max(m_A^i(t-1) + l_{\Phi_A^{(i)}}(t), 0), \quad (24)$$

$$n_A^i(\tau) = \sum_{k=0}^{i-1} \sum_{r=\tau-(k+1)D+1}^{\tau-kD} l_{\Phi_A^{(k)}}(r). \quad (25)$$

B. Performance Analysis

Similar to uncorrelated case, we have the following equation for node A

$$I_A \mathcal{D}_A + \sum_{n=1}^w I_{\Phi_A^{(n)}} [\mathcal{D}_A - nD]^+ \doteq \log \mathcal{F}_A, \quad (26)$$

where $I_{\Phi_A^{(n)}} = E_r [l_{\Phi_A^{(n)}}(r)]$.

It's easy to check (8) is a special case of the above equation. In the following, we further investigate a setting with one dimensional equally-spaced sensor array and joint Gaussian distribution concreteness. Suppose, $H_0 \sim N(\mathbf{u}, \Sigma)$ and $H_1 \sim N(0, \Sigma)$, namely both hypotheses are joint Gaussian distributions with different expectations and \mathbf{u} is a constant row vector $\mathbf{u} = [u, \dots, u]$. The correlation coefficient between two sensors is $\rho_{ij} = \rho^{|i-j|}$, where i, j are the corresponding indices of sensors. So, the covariance matrix Σ is given by:

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \rho & \dots & \rho^{(n+1)} \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \rho^{(n+1)} & \dots & \rho & 1 \end{bmatrix}. \quad (27)$$

Through asymptotic analysis of divergence, we have the following lemma, whose proof is given in Appendix I.

Proposition 1: for the scenario described above

$$\lim_{n \rightarrow \infty} \frac{1}{n} I_\Phi = \frac{u^2 \sigma^2}{2} \frac{1 - \rho}{1 + \rho}. \quad (28)$$

Prop. 1 gives the explicit relationship between the correlation coefficient and divergence. When correlation coefficient ρ is positive, the divergence is smaller with larger ρ . Therefore, highly correlated observations result in worse performance of multi-thread CUSUM test. This's easy to understand since less information is available with higher correlation. However, it is interesting to note that when ρ is negative, the divergence is larger with larger $|\rho|$, which actually improves the performance. Intuitively, negative correlation imposes a constraint that two observations lie on defferent sides with respect to the mean of a distribution; this renders two hypotheses easier to discriminate.

VI. NUMERICAL RESULTS

We now illustrate the analytical results in this paper via simulation. Note that the performance metric \mathcal{F} is not simulated since it is too large for a numerical simulation.

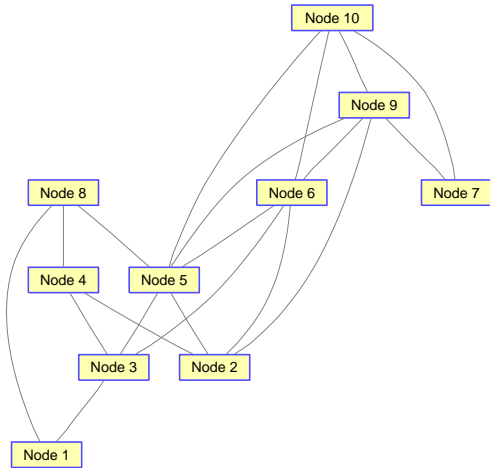


Fig. 4: Topology of network.

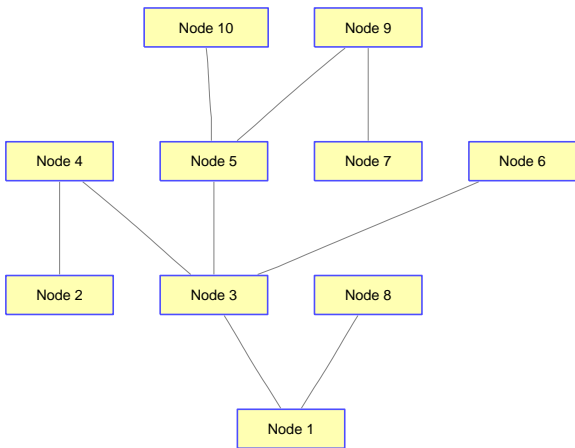


Fig. 5: Spanning tree generated by the algorithm in Subsection III-D.

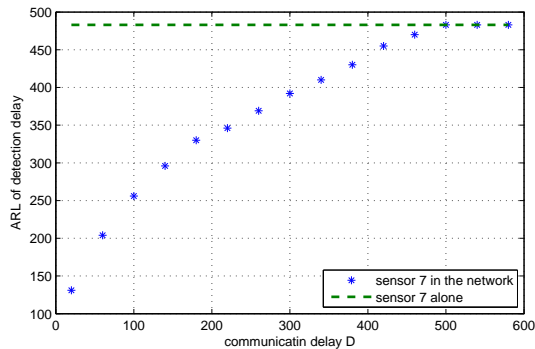
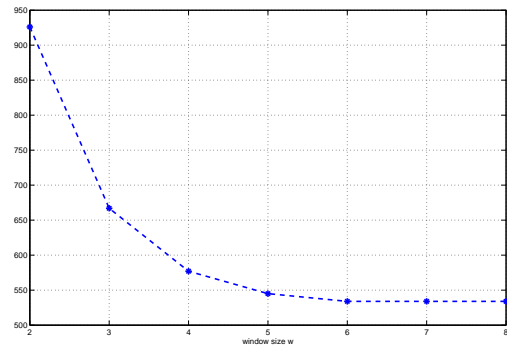
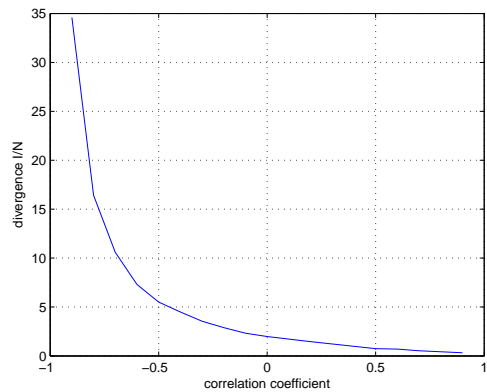


Fig. 6: The performance of node 7 in Fig. 5.

Fig. 7: Multi-node network: \mathcal{D} versus different window size w .Fig. 8: $\frac{1}{N}I_{\Theta}$ versus ρ .

A. Multi-node Network

Figure 4 shows the topology of a ten-node network. We assume that $H_0 \sim \mathcal{N}(-1, 1)$ and $H_1 \sim \mathcal{N}(1, 1)$ for all nodes. Fig. 5 is the spanning tree generated from the construction algorithm in Subsection III-D. We assume that $D = 100$ and $w = 4$. We observe that node 7 obtains the least aid from other nodes, whose performance is shown in Fig. 6 (note that, for different delays, we generate different spanning trees). Note that, even for the node with the least aid, the performance is still substantially improved when the communication delay D is small. Fig. 7 shows the maximum \mathcal{D} versus different window sizes w . We observe that larger window size implies better performance (since more observations are used for the quickest detection) while the performance gain increment is marginal when the window size is large.

B. Correlated Observations

Fig.8 shows the curve of $\frac{1}{N}I_{\Theta}$ as a function of correlation coefficient ρ for a ten-node one dimensional sensor array. We can see the simulation result is consistent with asymptotic analysis in Appendix I: divergence is increased with negative correlation coefficient.

VII. CONCLUSIONS

In this paper, we have made several extensions to the study on collaborative quickest detection in adhoc networks initiated in Part I of this two-part paper. In particular we have presented algorithms and performance analysis for a general N -node network, which also involves spanning tree generation and information exchange protocol. We have further expanded our model to include block transmission mode and correlated observations.

APPENDIX I PROOF OF PROP. 1

Proof: Similar to Proposition 3.1 of [15]. The covariance matrix Σ is a Toeplitz matrix, which can be decomposed as $\Sigma = \Psi\Lambda\Psi^T$, where Λ is a diagonal matrix composed of the eigenvalues $\{\lambda_k\}$ of Σ , and Ψ is a unitary matrix with eigenvectors $\{\psi_k\}$ of Σ .

Define $\mathbf{p} = \mathbf{u}\Psi$ and $\mathbf{w} = \mathbf{x}\Psi$, so $\{w_k\}$ are independent variables with distribution $\mathcal{N}(0, \lambda_k)$ under H_0 , and $\mathcal{N}(p_k, \lambda_k)$ under H_1 . Therefore, the log-likelihood ratio is

$$L_\Phi = \sum_{k=1}^n \left(\frac{w_k p_k}{\lambda_k} - \frac{p_k^2}{2\lambda_k} \right). \quad (29)$$

We define

$$\Omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{G(\omega)}{S(\omega)} d\omega, \quad (30)$$

where $G(\omega)$ is the power spectral of \mathbf{u} . Since \mathbf{u} is a constant, $G(\omega) = 2\pi u^2 \delta(\omega)$. $S(\omega)$ is spectral function of Σ [16], given by

$$S(\omega) = \sigma^2 \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos \omega}. \quad (31)$$

As a result

$$\begin{aligned} \Omega &= \int_0^{2\pi} \sigma^2 \frac{1 + \rho^2 - 2\rho \cos \omega}{1 - \rho^2} u^2 \delta(\omega) d\omega \\ &= u^2 \sigma^2 \frac{1 - \rho}{1 + \rho}. \end{aligned} \quad (32)$$

The logarithmic moment generating function of L_Φ is

$$\begin{aligned} \Delta_1(t) &= \log E_1 \left\{ \exp \left(t \left(\sum_{k=1}^n \frac{p_k w_k}{\lambda_k} - \frac{p_k^2}{2\lambda_k} \right) \right) \right\} \\ &= \sum_{k=1}^n \frac{(t - t^2) p_k^2}{2\lambda_k}. \end{aligned} \quad (33)$$

According to the properties of moment generating function

$$E_1(L_\Phi) = \frac{d}{dt} \Delta_1(t) \Big|_{t=0} = \sum_{k=1}^n \frac{p_k^2}{2\lambda_k}. \quad (34)$$

Base on Toeplitz Distribution Theorem [17],

$$\lim_{n \rightarrow \infty} \frac{1}{n} E_1(L_\Phi) = \lim_{n \rightarrow \infty} \frac{1}{n} I_\Phi = \frac{1}{2} \Omega = \frac{u^2 \sigma^2}{2} \frac{1 - \rho}{1 + \rho}. \quad (35)$$

REFERENCES

- [1] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes: Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [2] S. Ben-David, T. He and L. Tong, "Non-parametric approach to change detection and estimation in large scale sensor networks," *Proceedings of Conference on Information Science and Systems*, Princeton, NJ, March 2004.
- [3] R. W. Crow and S. C. Schwartz, "Quickest detection for sequential decentralized decision systems," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 10, pp. 267-283, Jan. 1996.
- [4] G. Lorden, "Procedures for reacting to a change in distributions," *Ann. Math. Statist.*, Vol. 42, pp. 1897-1908, 1971.
- [5] G. B. Moustakides, "Optimal stopping times for detecting changes in distributions," *Ann. Math. Statist.*, Vol. 14, pp. 1379-1387, 1986.
- [6] N. A. Nechval, "A general method for constructing automated procedure for quickest detection of a change in quality control," *Computers in Industry*, Vol. 10, pp. 177-183, July 1988.
- [7] E. S. Page, "Continuous inspection schemes," *Biometrika*, Vol. 41, pp. 100-115, 1954.
- [8] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer, New York, 1998.
- [9] T. Kailath and H. V. Poor, "Detection of stochastic processes," *IEEE Trans. on Inform. Theory*, Vol. 44, pp. 2230-2259, Oct. 1998.
- [10] D. Siegmund and E. S. Venkatraman, "Using the generalized likelihood ratio statistic for sequential detection of a change-point," *Ann. Math. Statist.*, Vol. 23, pp. 255-271, Aug. 1995.
- [11] R. J. Stahl and P. K. Willett, "Quickest detection of a tonal burst," *IEEE Trans. on Signal Processing*, Vol. 45, pp. 2037-2047, 1997.
- [12] V. V. Veeravalli, T. Basar and H. V. Poor, "Decentralized sequential detection with sensors performing sequential tests," *Mathematics of Control, Signals and Systems*, Vol. 7, pp. 292-305, 1994.
- [13] V. V. Veeravalli, "Decentralized quickest detection," *IEEE Trans. Inform. Theory*, Vol. 47, pp. 1657-1665, 2001.
- [14] V. V. Veeravalli and A. Tartakovsky, "Quickest change detection in sensor networks," *First International Workshop in Sequential Methodologies (IWSM)*, Auburn, AL, 2007.
- [15] W. Li and H. Dai, "Distributed detection in wireless sensor networks using a multiple access channel," *IEEE Transaction on Signal processing*, vol. 55, March 2007.
- [16] R. M. Gray, *Toeplitz and Circulant Matrices: A Review*, Stanford, CT: Free book, 2002.
- [17] U. Grenander and G. Szego, *Toeplitz and Forms and Their Applications*, New York; Chelsea, 1958.