

# Timely Group Updating

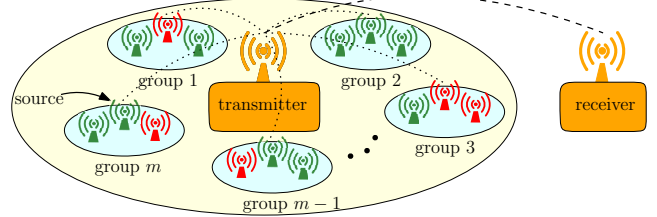
Melih Bastopcu      Sennur Ulukus  
 Department of Electrical and Computer Engineering  
 University of Maryland, College Park, MD 20742  
 bastopcu@umd.edu      ulukus@umd.edu

**Abstract**—We consider two closely related problems: anomaly detection in sensor networks and testing for infections in human populations. In both problems, we have  $n$  nodes (sensors, humans), and each node exhibits an event of interest (anomaly, infection) with probability  $p$ . We want to keep track of the anomaly/infection status of all nodes at a central location. We develop a *group updating* scheme, akin to group testing, which updates a central location about the status of each member of the population by appropriately grouping their individual status. Unlike group testing, which uses the expected number of tests as a metric, in group updating, we use the expected age of information at the central location as a metric. We determine the optimal group size to minimize the age of information. We show that, when  $p$  is small, the proposed group updating policy yields smaller age compared to a sequential updating policy.

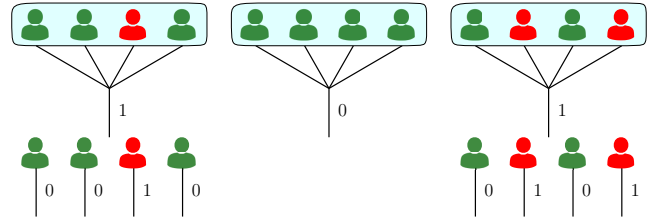
## I. INTRODUCTION

We consider two different problems with similar system models: anomaly detection in sensor networks and testing for infections in human populations. In the anomaly detection problem,  $n$  sensor nodes monitor a region and make measurements for an anomaly (e.g., fire, chemical spills, etc.) and report their measurements to a central location; see Fig. 1(a). Each sensor node detects an anomaly with probability  $p$  independent of others. In the infection testing problem, there are  $n$  individuals each of whom is infected with probability  $p$  independent of others, and their infection status needs to be tallied at a central location; see Fig. 1(b). In both problems, we want to identify the anomaly/infection status of each node as timely as possible in order to take necessary actions as quickly as possible, e.g., control the fire or isolate/treat the infected persons. For a measure of timeliness, we use age of information, which keeps track of the time elapsed since the last time the status of a node is updated.

Inspired by the *group testing* approach introduced in [1], we develop a *group updating* approach to maintain timely status updates at the central location. To that end, we divide  $n$  nodes into groups of  $k$  nodes each. In the case of anomaly detection, a local transmitter collects anomaly status of all nodes within the group. If there is no anomaly detected within the group, the local transmitter sends a single 0 to the central location. The central location, then, knows the status of all nodes within the group. On the other hand, if there is at least one anomaly detected within the group, the local transmitter sends a 1 to the central location. The central location, then, knows that there is at least one anomalous reading within the group. The



(a) An anomaly detection system with multiple sensor nodes. Sensors in red indicate an anomaly and sensors in green indicate no anomaly.



(b) An infection detection system in a human population. Persons in red are infected and persons in green are not infected.

Fig. 1. System models considered in this paper.

local transmitter then sends the individual measurements of the sensors (0s and 1s) to the central location one-by-one. Similarly, in the case of testing humans for infection, we divide  $n$  individuals into groups of  $k$  each. Within each group, we mix the test samples of the individuals and perform a single test. If the test result is a 0, we know that no one within the group is infected. If the test result is a 1, then, we know that at least one person within the group is infected. In the latter case, we test each person within the group individually one-by-one.

In the proposed group updating method, the group size  $k$  plays an important role in the performance of the system, i.e., in the resulting age. If  $k$  is too large, then the first update will likely result in a 1, and we will need to proceed to update the status of each node within the group one-by-one. This will increase the update duration, and hence, the age. On the other hand, if  $k$  is too small, then this will result in too many groups, and therefore, too many updates within an update cycle. This will increase the age as well. Thus, there is an optimum group size  $k$ , which is not too small, not too large. In this paper, we determine that optimum size for given  $n$  and  $p$ .

Age of information has been used as a measure of timeliness in many communication and networking scenarios, such as, in web crawling, social networks, queueing networks, caching systems, energy harvesting systems, scheduling in networks,

multi-hop multicast networks, lossless and lossy source coding, computation-intensive systems, vehicular, IoT, UAV systems, and so on [2]–[35]. With this paper, we are bringing age of information as a measure of timeliness to anomaly detection and testing for infections. Specifically relevant to our case is the setting of multi-source systems, where maximum age first (MAF) [22], maximum age difference (MAD) [23], Whittle index [24]–[26], slotted ALOHA with threshold [27], hierarchical cooperation [28] have been used to achieve good age performance. Different from most of these works, where only one source can be updated at a time, with the proposed group updating approach, we allow all sources in a group to be updated simultaneously with a single status update.

In this paper, we introduce a group updating approach where if all updates from the sources in the same group are 0, then the transmitter sends only a single status update representing the entire group, otherwise, the transmitter sends an update indicating that there is at least one 1 in the group, and proceeds to send all individual updates within the group one-by-one. For this updating method, for arbitrary  $n$  and  $p$ , we first find an analytical expression for the average age, which depends on the group size  $k$ . For given  $n$  and  $p$ , we find the optimal group size  $k$  that minimizes the age. Next, we compare the performance of the proposed group updating policy with the performances of the traditional scheduling methods, and observe that the proposed group updating policy achieves a lower age than the existing schemes when  $p$  is small. In addition, we compare the optimal group size  $k$  in the group updating problem here and in the group testing problem in [1] and observe that they are different in general indicating the difference of the metrics used.

## II. SYSTEM MODEL

We consider a system with  $n$  sources/nodes. We divide the  $n$  sources into groups of size  $k$ , where  $m = \frac{n}{k}$  is the number of groups. Without loss of generality, we assume that  $k$  divides  $n$ , and thus,  $m$  is an integer. We denote the status of the  $j$ th source in the  $i$ th group in the  $\ell$ th update cycle by  $X_{ij}(\ell)$ , where  $i = 1, \dots, m$ ,  $j = 1, \dots, k$ , and  $\ell \geq 1$ .  $X_{ij}(\ell)$  is an independent and identically distributed (i.i.d.) binary random variable for all  $i, j$  and  $\ell$ , with distribution,

$$X_{ij}(\ell) = \begin{cases} 1, & \text{with probability } p, \\ 0, & \text{with probability } 1 - p, \end{cases} \quad (1)$$

where a status 1 indicates an anomaly/infection, and a status 0 indicates no anomaly/infection.

Let  $S_{ij}(\ell)$  denote the service time for the status update of the  $j$ th source in the  $i$ th group in the  $\ell$ th update cycle. This is the time it takes for the status of the node to go through the system and be tallied at the central location. Note that if the status of all nodes in the  $i$ th group is 0, then the service time for all nodes in this group is equal to 1, as in this case, for the anomaly detection problem, the local transmitter needs to send a single 0 to convey the status of all nodes, and in the infection testing problem, a single test will determine the infection status of all nodes in the group. On the other hand, if

any one of the sources in the  $i$ th group generates 1 as a status update, the service time for the  $j$ th source in the  $i$ th group will be equal to  $j + 1$ , as in this case, an initial status update is sent representing the entire group,  $j - 1$  status updates are sent for the sources before source  $j$ , and a final update is sent for source  $j$  itself. Thus, the service time for the  $j$ th node in group  $i$  is a random variable with distribution,

$$S_{ij}(\ell) = \begin{cases} 1, & \text{with probability } (1 - p)^k, \\ j + 1, & \text{with probability } 1 - (1 - p)^k. \end{cases} \quad (2)$$

The service time of the entire  $i$ th group in the  $\ell$ th update cycle, denoted by  $W_i(\ell)$ , is equal to the service time of the last source in the  $i$ th group,

$$W_i(\ell) = S_{ik}(\ell), \quad i = 1, \dots, m. \quad (3)$$

As the central location wants to get timely updates from all sources, we track the age of each source at the central location separately. We denote the instantaneous age of source  $j$  in group  $i$  at time  $t$  by  $a_{ij}(t)$ , with  $a_{ij}(0) = 0$ . Age of each source at the central location increases linearly in time and drops to the age of the most recently received update once an update is received. The long term average age of node  $j$  in group  $i$  is given by,

$$\Delta_{ij} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T a_{ij}(t) dt. \quad (4)$$

The overall average age of all sources  $\Delta$  is equal to,

$$\Delta = \frac{1}{n} \sum_{i=1}^m \sum_{j=1}^k \Delta_{ij}. \quad (5)$$

Our aim is to find the optimal group size  $k^*$  that minimizes the average age of all sources  $\Delta$ , i.e.,

$$k^* = \arg \min_{\{k\}} \Delta. \quad (6)$$

In Section III, we first find the average age,  $\Delta$ , in (5).

## III. AVERAGE AGE ANALYSIS

With the group updating policy, the transmitter starts with sending updates from the sources in the first group. If all the updates from the first group are 0 (as shown with green balls in the first  $k$  lines in Fig. 2), then the transmitter sends a single 0 to update all the sources in the first group (that is why the delivery times of updates for all sources in the first group marked with arrows in Fig. 2 are equal to 1). After sending updates from the first group, the transmitter proceeds to send updates from the second group. If any one of the updates from the second group is equal to 1 (denoted by a red ball in the lines between lines  $k+1$  and  $2k$  in Fig. 2), then the transmitter first sends a 1 as a status update representing the entire group, and then sends individual updates from each source one-by-one. As shown in Fig. 2, the receiver gets the first update from the second group after 2 units of time. After sending updates from the second group, the transmitter proceeds to send updates from the third group, and so on, up until the

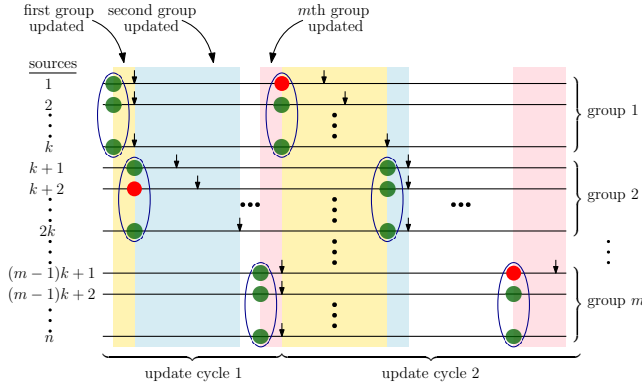


Fig. 2. A sample update generation and update delivery timeline. Lines 1 through  $n$  denote the nodes. Lines 1 through  $k$  denote the nodes in group 1. Green and red balls represent the anomaly/no anomaly status of each node. In update cycle 1, the yellow strip shows the time where the status of all nodes in group 1 is updated, the blue strip shows the time where the status of all nodes in group 2 is updated, and the pink strip shows the time where the status of all nodes in group  $m$  is updated. The process repeats itself in update cycle 2. Delivery times are marked by the downward arrows.

$m$ th (last) group. We call this entire time in which the status of all  $n$  sources are updated as update cycle 1 in Fig. 2. Once update cycle 1 ends, update cycle 2 starts all over again with all sources taking a new i.i.d. realization. In Fig. 2, in update cycle 1, the yellow vertical strip shows the time in which the status of all nodes in group 1 is updated, the blue strip shows the time in which the status of all nodes in group 2 is updated, so on so forth, and finally, the pink strip shows the time in which the status of all nodes in group  $m$  is updated.

Fig. 3 shows a sample age evolution curve for the  $j$ th source in the  $i$ th group at the central location, i.e.,  $a_{ij}(t)$ . Here,  $S_{ij}(\ell)$  defined in (2) denotes the service time of the  $j$ th source in the  $i$ th group in the  $\ell$ th update cycle. In addition,  $W_{ij}(\ell)$  denotes the total waiting time until the  $\ell$ th update is generated after the service completion of the  $(\ell - 1)$ th update for the same source. Thus,  $W_{ij}(\ell)$  is given by,

$$W_{ij}(\ell) = \bar{W}_{ij}(\ell - 1) + \sum_{r=i+1}^m W_r(\ell - 1) + \sum_{r=1}^{i-1} W_r(\ell), \quad (7)$$

where  $W_r(\ell)$  is given in (3), and  $\bar{W}_{ij}(\ell - 1)$  denotes the remaining service time of the  $i$ th group in the  $(\ell - 1)$ th update cycle which is given by  $\bar{W}_{ij}(\ell - 1) = W_i(\ell - 1) - S_{ij}(\ell - 1)$ .

We denote the length of the  $\ell$ th update cycle for the  $j$ th source in the  $i$ th group as  $Y_{ij}(\ell) = S_{ij}(\ell - 1) + W_{ij}(\ell)$  with  $S_{ij}(0) = 0$  for convention. One can show that the long term average age  $\Delta_{ij}$  given in (4) as in [5] is,

$$\Delta_{ij} = \lim_{N \rightarrow \infty} \frac{\frac{1}{N} \left( \frac{1}{2} \sum_{\ell=1}^{N+1} Y_{ij}(\ell)^2 + \sum_{\ell=1}^N Y_{ij}(\ell) S_{ij}(\ell) \right)}{\frac{1}{N} \sum_{\ell=1}^N Y_{ij}(\ell)}, \quad (8)$$

where  $N$  denotes the number of update cycles. We note that (8) can be written equivalently as,

$$\Delta_{ij} = \frac{\mathbb{E}[Y_{ij}^2]}{2\mathbb{E}[Y_{ij}]} + \mathbb{E}[S_{ij}]. \quad (9)$$

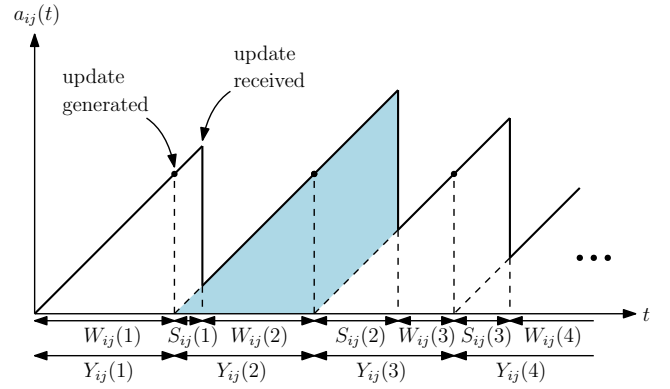


Fig. 3. A sample age evolution  $a_{ij}(t)$  at the central location.

We note that the length of an update cycle  $Y_{ij}$  is equal to the service completion time of all the groups, i.e.,

$$Y_{ij} = S_{ij} + W_{ij} = \sum_{r=1}^m W_r. \quad (10)$$

Therefore, the variable  $Y_{ij}$  does not depend on  $i$  or  $j$ . We thus denote  $Y_{ij}$  with a single random variable  $Y$ , i.e.,  $Y = Y_{ij}$ . On the other hand, from (2),  $S_{ij}$  depends on  $j$ , and we denote it by  $S_j$ . Then, the overall average age  $\Delta$  in (5) is equal to,

$$\Delta = \frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]} + \mathbb{E}[S], \quad (11)$$

where  $\mathbb{E}[S] = \frac{1}{n} \sum_{i=1}^m \sum_{j=1}^k \mathbb{E}[S_{ij}] = \frac{1}{k} \sum_{j=1}^k \mathbb{E}[S_j]$ .

Now, using (2)-(3), we have  $\mathbb{E}[W_r] = 1 + k(1 - (1 - p)^k)$  for all  $r$ . Thus, from (10),

$$\mathbb{E}[Y] = \frac{n}{k} + n(1 - (1 - p)^k). \quad (12)$$

In addition,

$$\begin{aligned} \mathbb{E}[Y^2] = & n(n - k)(1 - p)^{2k} + \frac{n^2(k + 1)^2}{k^2} \\ & - n \left( 2n \left( 1 + \frac{1}{k} \right) - k \right) (1 - p)^k. \end{aligned} \quad (13)$$

Further, from (2), we have,

$$\mathbb{E}[S_{ij}] = \mathbb{E}[S_j] = 1 + j(1 - (1 - p)^k), \quad (14)$$

and thus, we have,

$$\mathbb{E}[S] = 1 + \frac{k + 1}{2}(1 - (1 - p)^k). \quad (15)$$

Hence, the overall average age  $\Delta$  in (11) is

$$\begin{aligned} \Delta = & \frac{k^2(n - k)(1 - p)^{2k} + n(k + 1)^2}{2k + 2k^2(1 - (1 - p)^k)} \\ & - \frac{(2n(k + 1) - k^2)(1 - p)^k}{2 + 2k(1 - (1 - p)^k)} \\ & + 1 + \frac{k + 1}{2}(1 - (1 - p)^k). \end{aligned} \quad (16)$$

The overall average age in (16) depends on  $n$ ,  $p$  and  $k$ . We find the optimal  $k$  that minimizes  $\Delta$  numerically in Section V.

#### IV. GROUP UPDATING VERSUS GROUP TESTING

While the group testing and group updating policies are operationally similar, parameter selection, mainly selection of the group size in both problems, is different. In particular, in group testing, group size  $k$  is chosen to minimize the expected number of tests. In our terminology, expected number of tests corresponds to the expected length of an update cycle, i.e.,  $\mathbb{E}[Y]$ , as the transmitter sends one status update at a time. Thus, group testing chooses the group size  $k_{gt}^*$  by solving,

$$k_{gt}^* = \arg \min_{\{k \in \mathbb{Z}^+\}} \mathbb{E}[Y], \quad (17)$$

where  $\mathbb{E}[Y]$  is given in (12). In order for group testing to be more efficient than sequential updating of sources one-by-one, which uses  $n$  tests in an update cycle, we need  $\mathbb{E}[Y] \leq n$ , which implies  $p \leq p_{gt}$ , where

$$p_{gt} = 1 - \left(\frac{1}{k}\right)^{\frac{1}{k}}. \quad (18)$$

We note that  $p_{gt}$  attains its maximum value 0.3066 when  $k = 3$ . Thus, when  $p > 0.3066$ , group testing becomes inefficient compared to sequential updating of sources one-by-one.

Next, we find  $k_{gt}^*$  in (17) analytically. For that, we first relax the integer constraint on  $k$ . Then, by equating the derivative of  $\mathbb{E}[Y]$  in (12) with respect to  $k$  to zero, we obtain,

$$\frac{\partial \mathbb{E}[Y]}{\partial k} = -\frac{n}{k^2} - n(1-p)^k \log(1-p) = 0, \quad (19)$$

which gives,

$$\frac{k}{2} \log(1-p) e^{\frac{k}{2} \log(1-p)} = -\frac{1}{2} \sqrt{-\log(1-p)}. \quad (20)$$

Note that (20) is in the form of  $xe^x = y$ , whose solutions for  $x$  are  $x_1 = W_0(y)$  and  $x_2 = W_{-1}(y)$  when  $-\frac{1}{e} \leq y < 0$ . Here,  $W_0(\cdot)$  and  $W_{-1}(\cdot)$  denote the principle and  $-1$ st branches of the Lambert  $W$  function, respectively [36]. Thus, when  $0 < p \leq 1 - e^{-\frac{4}{n^2}} = 0.418$ , we have two solutions for (20) which are given by,

$$\alpha_1 = \frac{2}{\log(1-p)} W_0\left(-\frac{1}{2} \sqrt{-\log(1-p)}\right), \quad (21)$$

$$\alpha_2 = \frac{2}{\log(1-p)} W_{-1}\left(-\frac{1}{2} \sqrt{-\log(1-p)}\right). \quad (22)$$

When  $p > 0.418$ , one can show that  $\frac{\partial \mathbb{E}[Y]}{\partial k} < 0$ , and thus, the optimal  $k$  is equal to  $n$ . However, as the group testing method becomes inefficient when  $p > 0.3066$ , we only need to consider the case when  $0 < p \leq 0.418$ , and thus,  $\alpha_1$  in (21) and  $\alpha_2$  in (22) always exist.

Thus, in order to find the optimal  $k$ , we need to check  $k = \alpha_r^u$  where  $\alpha_r^u = \min\{k | k \geq \alpha_r, k | n\}$  for  $r = 1, 2$ ;  $k = \alpha_r^l$  where  $\alpha_r^l = \max\{k | k \leq \alpha_r, k | n\}$  for  $r = 1, 2$ ;  $k = 1$ ; and  $k = n$ . In other words, the optimal  $k$  is given by,

$$k_{gt}^* = \arg \min_{\{k \in \mathbb{K}\}} \mathbb{E}[Y], \quad (23)$$

where  $\mathbb{K} = \{1, \alpha_1^l, \alpha_2^l, \alpha_1^u, \alpha_2^u, n\}$ .

We perform a similar analysis for the group updating problem. Group updating chooses the group size  $k_{gu}^*$  by solving,

$$k_{gu}^* = \arg \min_{\{k \in \mathbb{Z}^+\}} \Delta, \quad (24)$$

where  $\Delta$  is given in (16). In order for group updating to be more efficient than sequential updating,  $\Delta$  in (16) needs to be smaller than  $\Delta_{\text{round-robin}}$ . For the round-robin (sequential) scheduling method,  $\mathbb{E}[Y] = n$ ,  $\mathbb{E}[Y^2] = n^2$ ,  $\mathbb{E}[S] = 1$ , and the overall average age from (11) is,

$$\Delta_{\text{round-robin}} = \frac{n}{2} + 1. \quad (25)$$

The condition  $\Delta \leq \Delta_{\text{round-robin}}$  gives an upper bound for the probability  $p$ , which we denote by  $p_{gu}$ . In other words, when  $p > p_{gu}$ , group updating becomes inefficient compared to sequential updating. Further, by relaxing the integer constraint on  $k$  and equating the derivative of  $\Delta$  in (16) with respect to  $k$  to 0, we can find the critical points where the age is minimized, and find  $k_{gu}^*$  analytically. Since  $\Delta$  in (16) is an involved function of  $k$ , in this work, we do not pursue analytical results on  $p_{gu}$  and  $k_{gu}^*$ . Instead, we find  $k_{gu}^*$  for given of  $p$  and  $n$ , and examine  $p_{gu}$ , numerically, in the next section.

#### V. NUMERICAL RESULTS

In this section, we provide four numerical results to illustrate the performance of the proposed group updating method, and also to show its difference from the group testing method. In all the numerical results, we only consider  $k$  values that divide  $n$ . For example, if  $n = 6$ , we consider  $k = 1, 2, 3, 6$ .

In the first numerical example, we compare the performance of the proposed group updating method with the performances of the existing updating policies of MAF and MAD. Since after receiving each update, the age at the receiver goes down to 1, MAF and MAD scheduling policies become identical. In addition, as the ages of all sources start from zero, MAF and MAD policies become the same as the round-robin scheduling method. The average age for the round-robin scheme is given in (25). We note that  $\Delta_{\text{round-robin}}$  increases linearly with  $n$  and does not depend on the probability  $p$ .

In the first numerical example, we take  $n = 120$  and plot in Fig. 4 the average age  $\Delta$  in (16) with respect to  $k$  when  $p = 0.01, 0.1, 0.2, 0.4$ , together with  $\Delta_{\text{round-robin}}$  in (25). We observe in Fig. 4 that, for all values of  $p$ , the average age first decreases with  $k$  and then increases with  $k$ , as initially, increasing  $k$  decreases the number of groups, making group updating more efficient, but after a while, further increasing  $k$  decreases the likelihood of having all zero updates in a group, requiring many follow-up individual updates. Thus, there is a trade-off between these two opposing factors, and there is an optimum group size to minimize the average age. As marked with a cross in Fig. 4, when  $p = 0.01$  the optimal group size is  $k_{gu}^* = 8$ ; when  $p = 0.1$  it is  $k_{gu}^* = 4$ ; when  $p = 0.2$  it is  $k_{gu}^* = 3$ ; and when  $p = 0.4$  it is  $k_{gu}^* = 3$ . We also observe that the group updating method becomes inefficient with increased  $p$  as it becomes more likely for the transmitter to send individual updates. When  $p$  is large enough, e.g., when

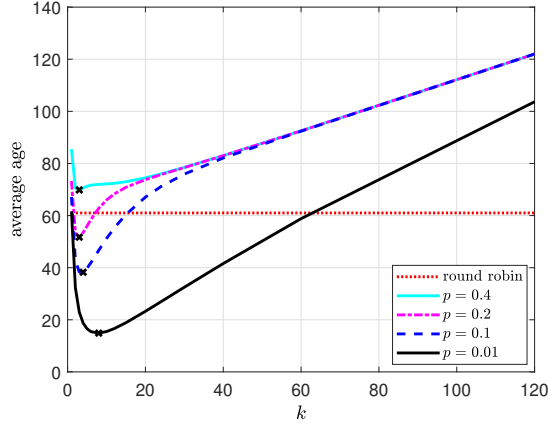


Fig. 4. Average age versus group size with the proposed group updating method and the round robin method when  $p = 0.01, 0.1, 0.2, 0.4$ .

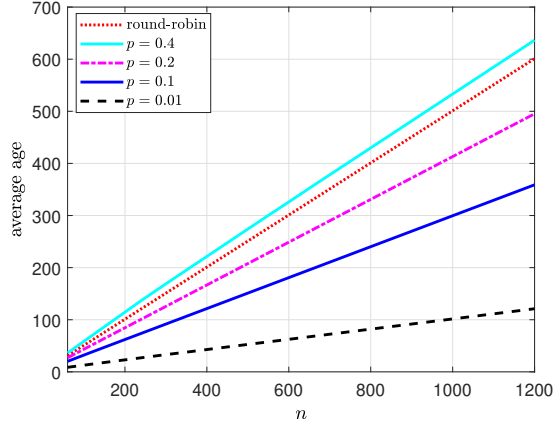
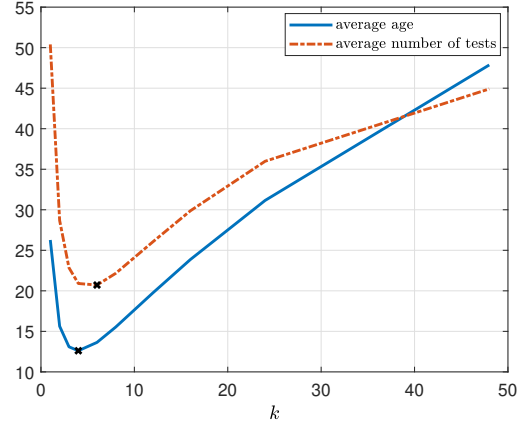


Fig. 5. Average age versus population size with the proposed group updating method and the round robin method when  $p = 0.01, 0.1, 0.2, 0.4$ .

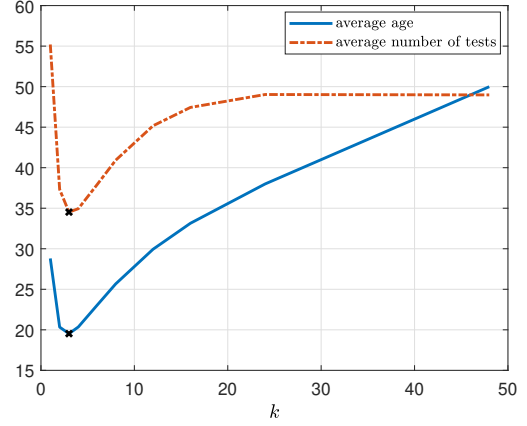
$p = 0.4$ , we observe in Fig. 4 that group updating becomes inefficient and does not improve the average age compared to the round-robin scheduling method.

In the second numerical example, we again take  $p = 0.01, 0.1, 0.2, 0.4$ , and plot in Fig. 5 the average age with respect to  $n$ , the population size, for  $n$  from 60 to 1200. For each value of  $p$  and  $n$ , we first find the optimal  $k_{gu}^*$  that achieves the minimum age, then plot that minimum age with respect to  $n$ . We observe in Fig. 5 that the average age increases linearly with the proposed group updating method as with the round-robin scheduling method. Similar to the first numerical example, the average age increases with  $p$  as group updating becomes less efficient with larger  $p$ .

In the third numerical example, we examine the differences between the group updating problem and the group testing problem. For this numerical example, we take  $n = 48$ ,  $p = 0.05, 0.15$ , and determine the optimal  $k$  values that minimize the average age and also the average number of updates. When  $p$  is small, e.g., when  $p = 0.05$ , we observe in Fig. 6(a) that the optimal group size that minimizes the average age is  $k_{gu}^* = 4$ , whereas the optimal group size that minimizes the average number of updates is  $k_{gt}^* = 6$ . This verifies that the group updating problem is different than the group testing problem.



(a)



(b)

Fig. 6. Average age for the group updating method and average number of updates for the group testing method with respect to  $k$  for  $n = 48$  when (a)  $p = 0.05$  and (b)  $p = 0.15$ .

However, when  $p$  is relatively large, e.g., when  $p = 0.15$ , we observe in Fig. 6(b) that the optimal group sizes in both problems are equal  $k_{gu}^* = k_{gt}^* = 3$ . In other words, when  $p$  gets larger, the optimal  $k$  values for the group updating and group testing problems get closer to each other.

In the fourth numerical example, we examine  $k_{gu}^*$  and  $k_{gt}^*$  as a function of  $p$ . We take  $n = 120$  and vary  $p$  between 0.01 and 0.25. We observe in Fig. 7 that both  $k_{gu}^*$  and  $k_{gt}^*$  decrease with probability  $p$ . With higher  $p$ , the sources in a group begin to generate more 1s as status updates, which results in sending more individual updates from the sources. Thus, decreasing the group size  $k^*$  in both of the problems helps counter the effects of increased  $p$ . Similar to the previous example, we observe in Fig. 7 that  $k_{gu}^*$  and  $k_{gt}^*$  are different when  $p$  is small, and become the same when  $p \geq 0.13$  for this choice of  $n$ .

## VI. CONCLUSION

We considered the problem of timely group updating, where similar to group testing, the sources are divided into groups; if all updates within a group are negative, a single group update suffices; if at least one update is positive, this triggers

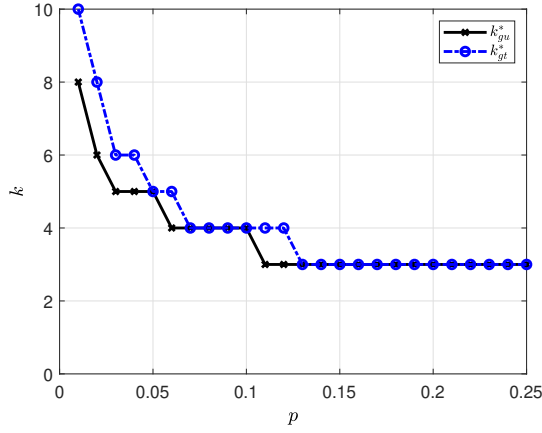


Fig. 7. Optimum group sizes  $k_{gu}^*$  in the group updating problem and  $k_{gt}^*$  in the group testing problem for  $n = 120$ , for  $p$  from 0.01 to 0.25.

a sequence of individual updates. For this updating scheme, we derived an analytical expression for the average age as a function of the group size  $k$ , the number of sources  $n$ , and the probability  $p$ . For given  $n$  and  $p$ , we found the optimal group size  $k$  that minimizes the age. We showed that when  $p$  is small, group updating performs better than sequential updating. We also showed that the optimal group sizes for group updating and group testing are different. This is because, while group testing aims to minimize the first moment of the length of an update cycle, group updating aims to minimize the age which depends on both the first and second moments of the length of an update cycle. An analogous observation was made in timely source coding versus traditional source coding, where the former depends on the first and second moments of the codeword length, while the latter depends only the first moment [30], [31].

## REFERENCES

- [1] R. Dorfman. The detection of defective members of large populations. *Ann. Math. Statist.*, 14(4):436–440, December 1943.
- [2] J. Cho and H. Garcia-Molina. Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems*, 28(4):390–426, December 2003.
- [3] S. Ioannidis, A. Chaintreau, and L. Massoulie. Optimal and scalable distribution of content updates over a mobile social network. In *IEEE Infocom*, April 2009.
- [4] M. Costa, M. Codreanu, and A. Ephremides. On the age of information in status update systems with packet management. *IEEE Transactions on Information Theory*, 62(4):1897–1910, April 2016.
- [5] E. Najm, R. D. Yates, and E. Soljanin. Status updates through M/G/1/1 queues with HARQ. In *IEEE ISIT*, June 2017.
- [6] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff. Update or wait: How to keep your data fresh. *IEEE Transactions on Information Theory*, 63(11):7492–7508, November 2017.
- [7] A. Soysal and S. Ulukus. Age of information in G/G/1/1 systems. In *Asilomar Conference*, November 2019.
- [8] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger. Age-optimal constrained cache updating. In *IEEE ISIT*, June 2017.
- [9] M. Bastopcu and S. Ulukus. Information freshness in cache updating systems. *IEEE Transactions on Wireless Communications*. To appear. Also available on arXiv:2004.09475.
- [10] M. Bastopcu and S. Ulukus. Maximizing information freshness in caching systems with limited cache storage capacity. In *Asilomar Conference*, November 2020.
- [11] S. Farazi, A. G. Klein, and D. R. Brown III. Average age of information for status update systems with an energy harvesting server. In *IEEE Infocom*, April 2018.
- [12] X. Wu, J. Yang, and J. Wu. Optimal status update for age of information minimization with an energy harvesting source. *IEEE Transactions on Green Communications and Networking*, 2(1):193–204, March 2018.
- [13] A. Baknina, O. Ozel, J. Yang, S. Ulukus, and A. Yener. Sending information through status updates. In *IEEE ISIT*, June 2018.
- [14] S. Leng and A. Yener. Age of information minimization for an energy harvesting cognitive radio. *IEEE Transactions on Cognitive Communications and Networking*, 5(2):427–439, June 2019.
- [15] A. Arafa and S. Ulukus. Timely updates in energy harvesting two-hop networks: Offline and online policies. *IEEE Transactions on Wireless Communications*, 18(8):4017–4030, August 2019.
- [16] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor. Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies. *IEEE Transactions on Information Theory*, 66(1):534–556, January 2020.
- [17] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas. A reinforcement learning framework for optimizing age of information in RF-powered communication systems. *IEEE Transactions on Communications*, 68(8):4747–4760, May 2020.
- [18] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis. The cost of delay in status updates and their value: Non-linear ageing. *IEEE Transactions on Communications*, 68(8):4905–4918, April 2020.
- [19] M. Bastopcu and S. Ulukus. Minimizing age of information with soft updates. *Journal of Communications and Networks*, 21(3):233–243, June 2019.
- [20] E. T. Ceran, D. Gunduz, and A. Gyorgy. A reinforcement learning approach to age of information in multi-user networks. In *IEEE PIMRC*, September 2018.
- [21] R. D. Yates and S. K. Kaul. The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3):1807–1827, March 2019.
- [22] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella. Age-optimal updates of multiple information flows. In *IEEE Infocom*, April 2018.
- [23] H. B. Beytur and E. Uysal-Biyikoglu. Minimizing age of information for multiple flows. In *IEEE BlackSeaCom*, pages 1–5, 2018.
- [24] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. Scheduling policies for minimizing age of information in broadcast wireless networks. *IEEE/ACM Transactions on Networking*, 26(6):2637–2650, December 2018.
- [25] Y. Hsu. Age of information: Whittle index for scheduling stochastic arrivals. In *IEEE ISIT*, June 2018.
- [26] A. Maatouk, S. Saad Kriouile, M. Assaad, and A. Ephremides. On the optimality of the Whittle’s index policy for minimizing the age of information. January 2020. Available on arXiv:2001.03096.
- [27] O. T. Yavascan and E. Uysal. Analysis of slotted ALOHA with an age threshold. July 2020. Available on arXiv:2007.09197.
- [28] B. Buyukates, A. Soysal, and S. Ulukus. Age of information scaling in large networks with hierarchical cooperation. In *IEEE Globecom*, December 2019.
- [29] B. Buyukates, A. Soysal, and S. Ulukus. Age of information in multihop multicast networks. *Journal of Communications and Networks*, 21(3):256–267, July 2019.
- [30] P. Mayekar, P. Parag, and H. Tyagi. Optimal source codes for timely updates. *IEEE Transactions on Information Theory*, 66(6):3714–3731, March 2020.
- [31] M. Bastopcu, B. Buyukates, and S. Ulukus. Selective encoding policies for maximizing information freshness. April 2020. Available on arXiv:2004.06091.
- [32] B. Buyukates and S. Ulukus. Timely distributed computation with stragglers. *IEEE Transactions on Communications*, 68(9):5273–5282, September 2020.
- [33] P. Zou, O. Ozel, and S. Subramaniam. Optimizing information freshness through computation-transmission tradeoff and queue management in edge computing. December 2019. Available on arXiv: 1912.02692.
- [34] M. Bastopcu and S. Ulukus. Age of information for updates with distortion. In *IEEE ITW*, August 2019.
- [35] M. A. Abd-Elmagid and H. S. Dhillon. Average peak age-of-information minimization in UAV-assisted IoT networks. *IEEE Transactions on Vehicular Technology*, 68(2):2003–2008, February 2019.
- [36] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, December 1996.