# Poster: Age of Information in Wireless Networks: from Theory to Implementation

Igor Kadota MIT LIDS kadota@mit.edu M. Shahir Rahman MIT LIDS shahir@mit.edu Eytan Modiano MIT LIDS modiano@mit.edu

# ABSTRACT

Emerging applications, such as smart factories and fleets of drones, increasingly rely on sharing time-sensitive information for monitoring and control. In such application domains, it is essential to keep information fresh, as outdated information loses its value and can lead to system failures and safety risks. The Age of Information (AoI) is a performance metric that captures how fresh the information is from the perspective of the destination. In this paper, we show that as the congestion in the wireless network increases, the AoI degrades sharply, leading to outdated information at the destination. Leveraging years of theoretical research, we propose and implement WiFresh: an unconventional architecture that achieves near optimal information freshness in wireless networks, regardless of the level of congestion. Our experimental results show that WiFresh can improve information freshness by two orders of magnitude when compared to an equivalent standard WiFi network.

### **ACM Reference Format:**

Igor Kadota, M. Shahir Rahman, and Eytan Modiano. 2020. Poster: Age of Information in Wireless Networks: from Theory to Implementation. In *The 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20), September 21–25, 2020, London, UK*. ACM, New York, NY, USA, 3 pages. https://doi.org/

# **1** INTRODUCTION

Emerging applications will increasingly rely on sharing timesensitive information for monitoring and control. Examples are abundant: monitoring mobile ground-robots in automated fulfillment warehouses at Alibaba and Amazon [8]; collision prevention applications for vehicles on the road

MobiCom'20, September 21-25, 2020, London, UK

© 2020 Association for Computing Machinery.



Figure 1: Software Defined Radio (SDR) testbed.

[4]; path planning, localization and motion control for multidrone formations [1]; real-time surveillance system using a fleet of ground-robots [5]; and data collection from sensors, drones and cameras for agriculture using the Azure Farm-Beats IoT platform [7]. In such application domains, it is essential to *keep information fresh*, as outdated information loses its value and can lead to *system failures* and *safety risks*.

The various time-sensitive applications implemented in [1, 4, 5, 7, 8], and many others, use the IEEE 802.11 standard (WiFi). WiFi is an attractive choice for it is low-cost, well-established, and immediately available in drones [1], computing platforms running Robot Operating System [5], and sensors [7]. Moreover, as showcased by these various implementations, *small-scale underloaded* WiFi networks are capable of supporting time-sensitive applications. A main shortcoming of WiFi<sup>1</sup> is congestion.

**Our contribution.** Leveraging years of theoretical research, we propose WiFresh: a network architecture that achieves near optimal information freshness, even when the wireless network is overloaded. We implement WiFresh at the MAC layer in the network of FPGA-based Software Defined Radios (SDRs) in Fig. 1 using hardware-level programming. To the best of our knowledge, this is the first AoI-based system implementation that goes beyond controlling the packet generation rate at the sources. In fact, as we will see, WiFresh eliminates the need for controlling the packet generation rates.

To illustrate the concept of *information freshness*, consider the monitoring system in Fig. 2 composed of a remote monitor, a wireless base station (BS) and N mobile sources. Each source  $i \in \{1, 2, \dots, N\}$  moves with an average velocity of  $v_i$  meters per second, generates status information from time to time, and sends this information to the remote monitor via the wireless base station. Status information can include

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM ISBN ...\$

https://doi.org/

<sup>&</sup>lt;sup>1</sup>WiFi or any other wireless technology employing First-Come First-Served (FCFS) queues and Random Medium Access Control mechanisms.

MobiCom'20, September 21-25, 2020, London, UK





the source's current position, inertial measurements and pictures of the environment. The remote monitor keeps track of the information, and is particularly interested in the position of the sources. Assume that packets are time-stamped. Naturally, the higher the time-stamp, the fresher is the information contained in a packet. Let  $\tau_i(t)$  be the time-stamp of the freshest packet received by the destination from source *i by time t*. Then, we define the Age of Information (AoI) as  $h_i(t) := t - \tau_i(t)$ . The AoI is a recently proposed performance metric [3] that captures how fresh the information is from the perspective of the destination. The value of  $h_i(t)$  increases linearly in time while no fresher packet from source *i* is received, representing the information getting older. At the moment a *fresher* packet is received by the destination, the value of  $\tau_i(t)$  is updated and  $h_i(t)$  decreases to the packet *delay.* An AoI of  $h_i(t) = 2$  seconds represents that at time t the remote monitor knows the location of source i two seconds ago. Hence, the uncertainty about source i's position *at time t* is captured by the quantity  $v_i h_i(t)$ , as illustrated in Fig. 2, and a large AoI corresponds to a large uncertainty.

To capture the freshness of the information in the network, we define the expected network AoI (NAoI) as

$$\lim_{T \to \infty} \frac{1}{TN} \int_{t=0}^{I} \sum_{i=1}^{N} \mathbb{E}\left[h_i(t)\right] dt , \qquad (1)$$

where T > 0 is the time-horizon. To minimize the expected NAoI, we consider the wireless network as a whole and optimize the system across the queueing discipline, the multiple access mechanism and the transmission scheduling policy.

#### 2 **DESIGN AND IMPLEMENTATION**

The superior performance of WiFresh in terms of information freshness is due to the combination of three elements:

- Polling Multiple Access mechanism that prevents packet collisions, allowing for efficient resource allocation among sources, which is critical in congested networks and in networks with large number of sources N;
- Max-Weight (MW) policy that determines the sequence of sources to poll in order to minimize NAoI, keeping the information from every source as fresh as possible; and
- Last-Come First-Served (LCFS) queues that prioritize the *packet with lowest delay*, leading to sources that always transmit the freshest packets to the destination.

The choice of each of these elements is underpinned by theoretical research. In [2] and other works, the LCFS queue was shown to be the optimal queueing discipline for AoI. In our

prior work [6, Chapter 3], the Polling mechanism with MW policy was analyzed in different settings. Notice that Polling is needed to support the MW policy.

Bridging theory and practice. Theoretical works on AoI often assume that: 1) sources in the network are synchronized; 2) sources generate packets on-demand or according to known stochastic processes; 3) channel reliabilities associated with the wireless links  $\{p_i\}_{i=1}^N$  are static and known; and/or 4) feedback is instantaneous and without error. To leverage the theory and implement (for the first time) an AoI-based network architecture composed of LCFS queues and Polling mechanism with MW policy, we augment WiFresh with algorithms that synchronize clocks, dynamically learn  $\{p_i\}_{i=1}^N$ , and recover from transmission errors. Notice that clock synchronization is needed to accurately compute  $h_i(t) := t - \tau_i(t)$ , where *t* is the current time *mea*sured by the BS and  $\tau_i(t)$  is a time-stamp created by source i. If clocks are not synchronized, the values of  $h_i(t)$  for different sources may have different biases.

WiFresh source. The source generates information updates at the Application layer and forwards them to lower layers of the communication system. When a data packet arrives at the MAC layer, WiFresh appends a time-stamp to the packet and then stores it in a LCFS queue. The source can be in one of two states: 1) waiting for a poll packet from the BS; or 2) transmitting its freshest data packet to the BS. Upon receiving a poll packet, if the queue is empty, the source transmits an *empty packet* to the BS. The empty packet is used by the BS to differentiate between not receiving data due to a transmission error or due to an empty queue at the source, which impacts the estimation of the channel reliabilities  $\{p_i\}_{i=1}^N$  at the BS.

WiFresh Base Station. The BS does not generate data packets. Its main responsibility is to coordinate the communication in the network. The BS can be in one of two states: 1) waiting for a data packet; or 2) transmitting a poll packet. While waiting for a data packet, the BS keeps track of the waiting period. If the waiting period exceeds 100  $\mu$ sec or a data packet is received, the BS updates its estimate of the network state  $(\hat{h}_i(t), \hat{p}_i(t))_{i=1}^N$ , where  $\hat{h}_i(t)$  is the current AoI  $h_i(t)$  with time-stamp offset correction from the clock synchronization mechanism, and  $\hat{p}_i(t)$  is the estimate of the channel reliability associated with source *i*. These estimates are used by the MW policy to determine the next source to poll. In particular, at the decision time t of the next poll packet, the MW policy selects the source  $i^*(t)$  with highest index  $I(i, t) = \hat{p}_i(t)\hat{h}_i^2(t)$ . After transmitting a poll packet to source  $i^*(t)$ , the BS goes back to waiting for a data packet.

Implementation. WiFresh is implemented at the MAC layer using a network of FPGA-based SDRs composed of ten sources and a wireless BS, as shown in Fig. 1. To achieve Aol in Wireless Networks: from Theory to Implementation



Figure 3: NAoI (in milliseconds) for the testbed with ten sources generating packets of 150 bytes with rate  $\lambda$  (in Hz).

high performance, the LCFS queues, the Polling mechanism, and the MW policy were *fully implemented in FPGAs* with 10 MHz clocks, enabling WiFresh to make the scheduling decision and trigger the transmission of the next poll packet in approximately 20  $\mu$ sec. Keeping this time-interval short and limiting the length of the poll packet are important factors in reducing the control overhead and achieving high performance. The code is developed based on a modifiable WiFi reference design. The PHY layer of the reference design, which is kept unchanged, is based on the IEEE 802.11n with center frequency 2.437 GHz and 20 MHz bandwidth.

# **3 EXPERIMENTAL RESULTS**

In this section, we evaluate the performance of WiFresh in a dynamic indoor office space with *multiple external sources of interference* such as mobile phones, laptops and campus WiFi base stations. In particular, we evaluate and compare:

- WiFresh: as described in §2;
- WiFresh FCFS: WiFresh with sources employing FCFS;
- WiFi: UDP over standard WiFi; and
- WiFi LCFS: WiFi with sources employing LCFS.

We consider the SDR network in Fig. 1 with ten sources, each generating packets of 150 bytes with rate  $\lambda$ . These short packets of 150 bytes represent status updates, and different values of  $\lambda$  represent different levels of congestion. In Fig. 3, we display the expected NAoI measurements in milliseconds. Each experiment runs for 10 minutes.

**Measurements.** By comparing the results of WiFresh and WiFi for  $\lambda \ge 500$  Hz, we can see that *WiFresh improves information freshness by (at least) a factor of 200 when compared to an equivalent standard WiFi network*. To understand how much of this improvement is due to the queueing discipline and how much is due to the multiple access mechanism, we draw additional comparisons. By comparing WiFresh and WiFi LCFS, both of which use LCFS queues, we can assess the impact of the multiple access mechanism on NAoI. As expected, the improvement of Polling over Random Access increases with the network congestion. In particular, for  $\lambda = 5$  kHz, WiFresh improves NAoI by a factor of 7 when compared to WiFi LCFS. To assess the impact of queueing, we compare WiFresh and WiFresh FCFS, both of which use Polling

with MW policy. For  $\lambda \ge 500$  Hz, the LCFS queue improves information freshness by (at least) a factor of 100 when compared to the FCFS queue. In summary, both the queueing discipline and the multiple access mechanism improve NAoI significantly, but the effect of queueing is dominant.

**Congestion control.** The results in Fig. 3 show that the combination of LCFS queues and Polling mechanism with MW policy is the *only in which a higher rate*  $\lambda$  (*always*) *leads to a lower NAoI*, meaning that the WiFresh architecture eliminates the need for controlling the packet generation rate at the sources. Notice that any of the other three architectures, which employ either FCFS queues or Random Access, need to control  $\lambda$  in order to minimize NAoI.

**Applications of WiFresh.** From the measurements in Fig. 3, it is clear that the more congested the network, the more prominent is the superiority of WiFresh when compared with WiFi in terms of information freshness, making *WiFresh well-suited for large-scale applications that rely on sharing large amounts of time-sensitive information*. Examples of such applications are [1, 4, 5, 7, 8].

**Future work.** A challenge of implementing WiFresh at the MAC layer is the complexity associated with hardwarelevel programming. We plan to design an alternative implementation of WiFresh that runs at the Application layer, making it easy to integrate into time-sensitive applications that are implemented over WiFi such as [1, 4, 5, 7, 8].

## ACKNOWLEDGMENTS

This work was supported by NSF Grants AST-1547331, CNS-1713725, and CNS-1701964, and by Army Research Office (ARO) grant number W911NF-17-1-0508.

## REFERENCES

- J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus. 2019. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots* (2019).
- [2] M. Costa, M. Codreanu, and A. Ephremides. 2016. On the Age of Information in Status Update Systems with Packet Management. *IEEE Transactions on Information Theory* (2016).
- [3] S. Kaul, R. Yates, and M. Gruteser. 2012. Real-Time Status: How Often Should One Update?. In Proc. IEEE INFOCOM.
- [4] M. Klapez, C. A. Grazia, and M. Casoni. 2020. Application-Level Performance of IEEE 802.11p in Safety-Related V2X Field Trials. *IEEE Internet* of Things Journal (2020).
- [5] L. Matignon and O. Simonin. 2018. Multi-Robot Simultaneous Coverage and Mapping of Complex Scene - Comparison of Different Strategies. In *Proc. AAMAS*.
- [6] Y. Sun, I. Kadota, R. Talak, and E. Modiano. 2019. Age of Information: A New Metric for Information Freshness. Morgan & Claypool.
- [7] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman. 2017. FarmBeats: An IoT Platform for Data-Driven Agriculture. In *Proc. NSDI*.
- [8] P. Wurman, R. D'Andrea, and M. Mountz. 2007. Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. In Proc. IAAI.