2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Impact of message sorting on access to novel information in networks

Benjamin D. Horne & Sibel Adalı
Rensselaer Polytechnic Institute
Troy, NY 12180
Email: {horneb,adalis}@rpi.edu

Kevin Chan
US Army Research Laboratory
Adelphi, MD 20783
Email: kevin.s.chan.civ@mail.mil

***Abstract*—In social networks, individuals and systems work side by side. While individuals make decisions to filter or forward information, systems also prioritize and sort information to manage and assist individual information processing. It has long been argued that system level manipulations can reduce access of individuals to novel information. In this paper, we study how sorting of messages in one's inbox can help or hinder access of diverse information in the network through simulation of cognitively bounded actors. We show that first-in-first-out (FIFO) method of message sorting is ideal in bursty information arrival rates and in networks with lower diameter. Last-in-first-out (LIFO) method of message sorting is ideal for streaming information arrival, but leads to information overload in bursty scenarios by creating too many redundant copies of some of the information in the network. In short, the ideal message sorting method that enhances access to diverse information depends on the network type and information access patterns.**

I. Introduction

We live in the age of information overload. Just as Dabbish and Kraut define email overload [1], information overload is when "users perceive that their own use of social or communication platforms has gotten out of control because they receive and send more information than they can handle, find, or process effectively." Users increasingly use social networks for more than just communicating with peers, but for gaining knowledge of current events, expert opinion, political engagement, and social movements. As a result, information becomes available from an increasing number of networked sources instead of a few reputable news sources that were typically used in the past. This leads to increased duplication of information within the network, making it more difficult to gain access to unique and diverse pieces of information. This problem is complicated further by algorithms that filter and rank information based on different assumptions and motivations. For example, an ongoing debate is whether the Facebook algorithms that rank content based on user's estimated interests and strength of their friendship ties hurts access to diverse content, which has been said to be associated with "adopting more extreme attitudes over time and misperceiving facts about current events" [2]. This idea of the social harm caused by homophily based algorithms has gained more than just niche academic attention, but also some mainstream attention, showing up on podcasts such as NPR's "Note to Self [3]." In particular, "Note to Self" points out that many users are purposefully not engaging with ideologically similar information in an attempt to diversify their newsfeeds. Ultimately, the choice of algorithm to rank content is likely strongly dependent on what specific types of information a given site is used for, but in general the diversity of valuable content is desired. Do these relevence algorithms improve or harm user's access to information? More generally, how do algorithms to sort information impact access to information?

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA

To study this problem, we consider an idealized scenario in which a set of cognitively bounded actors use the same information system which sorts the messages received by them. Actors process the information as it is sorted and share with other actors in the network. We track how widely information is disseminated in the network. We find big differences between first-in-first-out (FIFO) and last-in-first-out (LIFO) message sorting schemes. While FIFO is ideal in events creating a burst of messages and LIFO is ideal in situations where information becomes available over a longer time period. In networks with smaller diameter, access to diverse information improves for FIFO sorting, while LIFO is generally uneffected. Our findings support an adaptive strategy for inbox management: using LIFO in most cases but shifting to FIFO in times of high information load especially within networks with low diameter.

II. Related Work

Rodriguez, Gummadi, and Schoelkopf [4] investigate the notion of information overload as it pertains to social contagions on Twitter, finding evidence of user information processing capacity thresholds on incoming and outgoing information. This work illustrates that when users hit this information overload threshold, they begin to prioritize information sources rather than simply using the LIFO feed of tweets. Also, empirical studies [4] illustrate that "background traffic" has an impact on "users' queueing delays"; defined as "the time users take to process information they receive." In contrast with this work, we do not directly investigate information overload, but the impact of message sorting on information propagation in situations involving information overload.

A common concern in networks is whether individuals are exposed to diverse sets of ideas [2] or whether individuals get news that agree with their ideological view in online social networks like Facebook [5]. Tufekci [6] points out the need to deeply consider what things we “turn over” to an algorithm and how that consideration can change how we think about algorithims like a news feed sorting algorithm. Along these same lines, Eslami *et al* developed a system that compares Facebook’s algorithmically curated news feed to a unsorted news feed [7]. Our approach is more general as we concentrate on the direct impact of message sorting in the dissemination of information in the network. We do not study ideology based filtering or homophily based sorting of methods. These are future extensions of our work, building on the foundation of this paper.

III. Problem Setting

In this paper, we consider a model in which a set of actors share information with each other along undirected network ties. We assume actors obtain information, called factoids (or messages) in our model, from outside sources and share it with their neighbors in the network. For simplicity, we assume each new factoid is known to only one actor initially. Actors disseminate information by processing them first and copying it to their neighbors. Factoids are sent verbatim, similar to a rewteet. We track the number of unique factoids actors come to know as a result of this process. The more factoids the actors have seen, the better informed they are. Our aim is to find conditions that improve or hinder the dissemination of novel and diverse information in the network.

a) Actor characteristics: Actors in the simulation have two mutable attributes: capacity and propensity to send, which are both fixed throughout the simulation. *Capacity (C)* is the number of factoids an actor processes at any one simulation step, which is set to 1 in this paper modeling cognitively bounded actors.

For each factoid the actor processes, she decides randomly whether to send it or not, depending on her *propensity to send (p_s)*. If a factoid is sent, it is copied to all the neighbors. The information received from different actors arrive in random order removing the possibility of bias towards a specific actor. Same factoid may be received multiple times from other actors and reprocessed. However, an actor will not send the same information twice.

b) Inbox sorting: Actors use a software system that allows them to organize their messages. All factoids that are yet to be processed are put in their inbox which are are sorted using one of the two basic methods: (i) *LIFO (last in first out)* sorts messages by the reverse order they are received in, the latest message on top, similar to Twitter. (ii) *FIFO (first in first out)* sorts messages in the order they are received in, the oldest message on top. This is a common option in many email applications as well as LIFO.

c) Simulation Model and Performance Measures: Each simulation runs a fixed number of steps. Actors are connected through a network of ties which are fixed throughout the simulation. If information is available at the beginning of the simulation, it is distributed to the actors randomly. At other simulation steps, actors may obtain new information from outside of their network and share it within the network before other information in their inbox. In other words, agents process what is in their inbox normally, unless their attention is shifted to new information. This design choice is intuitive as real-life social network users process and share information they find outside of the network before processing other information in the newsfeed. Basically, a user would not “hold on” to a news story she wants to share until she scrolls through and processes her newsfeed.

At each step of the simulation, all actors will act at the same time, removing a factoid from their inbox and decide whether to send it or not. After all actors have finished their send actions, sent factoids are placed in the inboxes of the recipients. Given this model, we track a number of performance measures at each simulation step such as the total number of distinct factoids known by the agent and the total number of agetns sending information at any point in time.

d) Setup: In this work, we use the Watts-Strogatz Small World model to generate the underlying network structure as it is easier to vary diameter and modularity while holding density fixed. All networks have 256 nodes and the same density (10 neighbors per node in average), all actors will have capacity 1 and 0.6 propensity to send, unless stated otherwise.

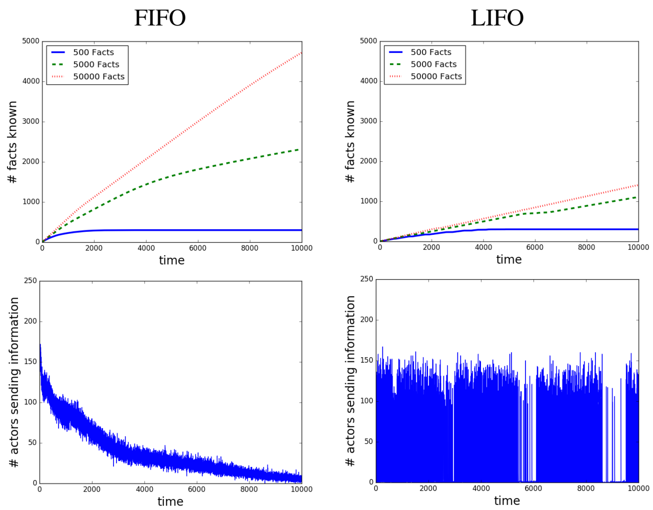
We consider two main scenarios for arrival pattern of information as shown in Table I. Total number of messages at the end of simulation is identical in both cases.

In the burst scenario, all factoids arrive in the beginning of the simulation and are distributed randomly to the actors. This models a case of intense activity and attention by the actors, such as after a major event. There are lot of different pieces of information and actors are interested in learning as much as possible from the network.

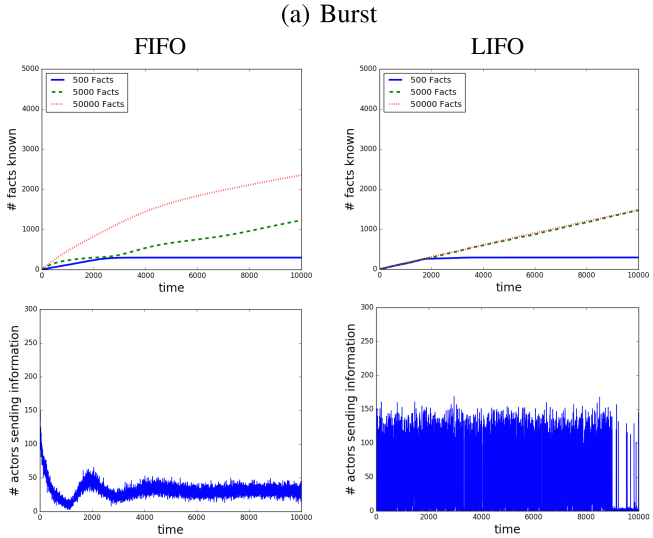
The second scenario involves streaming data, in which new information enters the network through all the actors in the network at regular time intervals. The actors start with some small set of factoids already in their inbox, then randomly generate small set of factoids over time. This second scenario models a period of steady activity, actors process information from network or bring new information throughout a period of time. Unless otherwise stated, simulations will involve 5000 messages (factoids) with a single copy of each. With a propensity to send of 0.6 and 5000 factoids, we expect about 3000 messages to circulate in the network in the best possible case.

Type	Details
Burst	5000 messages in a single burst at the beginning of simulation
Stream	5000 messages streamed at 1 message per 5 simulation steps, with 500 messages randomly distributed in the beginning of simulation

TABLE I
Information arrival patterns considered in this paper.



(a) Burst



(b) Streaming

Fig. 1. Number of facts known by an actor for different amount of information in the network and number of actors sending information for 5000 factoids in the network over time for burst (top) and stream (bottom) information arrival types using FIFO/LIFO inbox sorting methods.

IV. RESULTS

In this section, we illustrate the various trade offs of different information arrival and inbox sorting patterns, along with the impact of network structure. Each experiment is done 50 times and the results are averaged.

1) Basic Sorting, Burst Arrival: First, we explore how LIFO and FIFO inbox sorting methods impact access to diverse information in networks. In these experiments, we compare LIFO and FIFO sorting using the burst information arrival setting described in Table I, but vary the number of factoids between 500, 5,000 and 50,000.

As shown in Figure 1 (a), LIFO and FIFO perform roughly the same in disseminating information when there is very little information in the network (relative to its size). However, as the amount of available information in the system increases, LIFO sorting struggles to disseminate unique information, while FIFO sorting continues to perform reasonably well, scaling with the amount of information in the system.

This difference is caused by the oscillation of sending and filtering actions performed by actors. Recall that actors will not send information that they have already sent. As a result, when actors receive a duplicate of a factoid that they have sent before, they will use their time to simply remove this factoid from their inbox and not send anything. In LIFO, recently seen factoids are always on top of the inbox. This increases the likelihood of receiving a duplicate of a factoid shortly after sending it. In fact, many agents may be processing copies of a the same small set of factoids, resulting in a type of *synchronization*.

This type synchronization is akin to a large body of previous work on synchronization in complex networks [8] [9]. In particular, this synchronization is similar to epidemic flare-ups, in which people become ill and recover around the same times [10]. The key difference between the type of synchronization in the complex network literature and our work is the queueing of states. Rather than synchronizing on a single state or oscillating between different states, we are sychronizing on varing sets of states, in our case factoids, in a sorted queue. The large number of possible states makes this synchronization behavior hard to visualize. Instead, we look at when actors are sending information (i.e. they have received a new factoid) and when they are filtering information (i.e. they must likely have a duplicate in their inbox). The graphs in the bottom of Figure 1 (a) show this behavior. Notice that LIFO is constantly jumping between all of the actors filtering (zero actors sending) to majority of the network sending. This time spent filtering prevents new information from being sent and accessed.

In contrast, FIFO does not suffer from this synchronization behavior. In the FIFO setting, actors are sending information throughout the simulation, only reaching zero actors sending when all of the information in the system has been processed. For the LIFO setting to reach the point of all factoids being processed, it will take much longer. Conceivably, we can think of these two very different performances like springs, in which the FIFO setting is a critically damped spring and the LIFO setting is an almost completely undamped spring. This analogy is certainly not perfect, but provides a high level way of thinking about the differences between the two basic sorting methods in a high traffic network.

2) Burst vs. Streaming Information Arrival: We now consider streaming information arrival as shown in Figures 1 (b). Streaming information balances out the difference in LIFO and FIFO sorting performance, as FIFO performance worsens and LIFO performance improves. This result can be explained as the breaking up of synchronization in LIFO as new information being placed on top of an actor's inbox *breaks* her synchronization on a set of already sent or duplicate factoids. This break in-turn provides new information to the actor's neighbors, effectively breaking up their synchronization as well.

In the FIFO setting, information travels along paths in the network until it is completely filtered out. The longer it stays in the network, the more likely it will reach all the actors. Since, FIFO sorting can get unique information out to users very quickly, many actors will find their inboxes empty for some time while waiting for new factoids to arrive. This waiting creates small waves of sending and waiting. Hence, by the end of a 10,000 step simulation, actors will have done fewer send actions resulting in less information being shared.

It is important to note, the comparison between the burst and stream information arrival patterns is not direct. In the burst setting, all messages have the same amount of time in the system, while in the stream setting, certain messages will have a longer time in the system than others. For example, if a new message from the outside enters the network at step 9800 of 10,000 total steps, it will only have 200 steps to be diffused. In contrast a message that enters the system at the very beginning will have all 10,000 steps to be diffused. Thus, these two arrival patterns should be thought of as exclusive network scenerios: a high traffic time and a normal low traffic time. Both settings lend themselves to real-life behavior in social networks, but are not directly comparable.

3) Single Message Branching Factor: To further illustrate the difference between LIFO and FIFO inbox sorting methods, we view information cascades as branching processes. Specifically, we define the "branching factor" of a process as the average number of copies that an agent generates for any message received in her inbox. This metric is calculated for each message by counting the number of copies of that message generated by every agent. If an agent forwards a factoid, the count will be increased by the degree of that actor, since agents broadcast messages in our model. If an agent chooses not to forward a factoid, nothing will be added to the count. This method for calculating the branching factor is borrowed from [15]

In the burst setting, we note that LIFO proportionally creates a large number of copies of some factoids (making them viral), while the remaining factoids have very few copies. Of course, in LIFO there are many more factoids with zero copies as shown in Figure 1, but these are excluded in our analysis. In other words, a higher proportion of factoids are forwarded by many or almost all of the agents. FIFO on the other hand creates a more diverse set of branching factors, but causes less messages to go viral.

In the stream setting, both LIFO and FIFO's average branching factors are decreased due to actors doing fewer send actions as described in section IV-2. LIFO still creates many viral branching factors, but also slightly decreases the number of facts that are copied zero times. This decrease in messages copied zero times is not shown in the averages displayed in Table II as messages sent zero times are excluded. The branching factor distributions can be seen in Figure 2. In Figure 2, we also exclude facts that were sent zero times as we are interested in the facts that have been distributed.

In general, the branching factor of a process can be analytically estimated. For FIFO sorting, this estimation is fairly straight forward. Consider a single factoid f in the system starting at agent a. Assume agents are connected through a tree structure, abstracting out the possibility of cycle connections [15]. Agent a has a probabilty p_s to forward f to her k friends. Each of those k friends independently has probability p_s of forwarding f to each of his or her k friends. By independence, this pattern continues with decreasing probability as the depth of the cascade increases. Given enough time in the simulation relative to the number of incoming factoids, all factoids will be seen even in high traffic times.

Setting	Average Branching Factor	Maximum Branching Factor
LIFO Burst	2545	2560
FIFO Burst	1991	2560
LIFO Stream	2498	2560
FIFO Stream	1918	2560

TABLE II
AVERAGE AND MAX BRANCHING FACTORS FOR EACH SORTING AND INFORMATION ARRIVAL SETTING EXCLUDING MESSAGES THAT WERE COPIED 0 TIMES

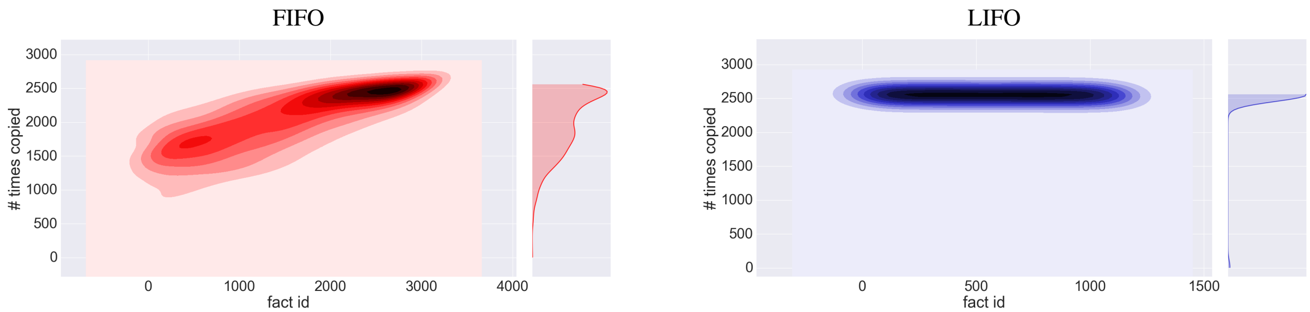
For LIFO sorting, this estimation is slightly more complex. For the sake of brevity, we will simply refer to [15] for the LIFO branching factor analysis. It has been shown that the number of copies is generally exponentially distributed in scenarios involving cognitively bounded actors (i.e. limited number of actions at each time step). Furthermore, the probability a message becoming "viral" increases as actors' propensity to send increases.

A. Network Connectivity

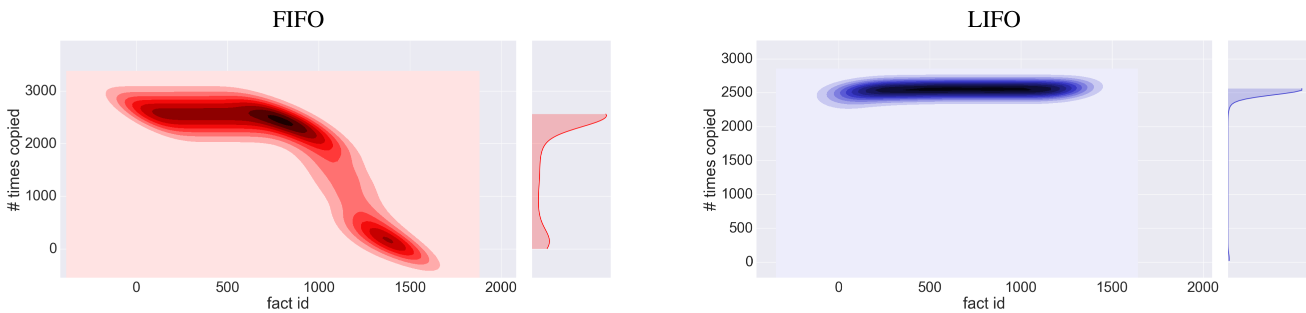
Up to this point, we have not discussed the impact of the underlying network of ties on information spread. To study this problem, we run simulations on diameter-varying small world networks using 2500 factoids in the system. Small world networks are parameterized by a single "rewiring" probability p_c. As p_c increases, more "short-cuts" are created, which will shrink the diameter of the graph and introduce small world behavior, resulting in a completely random graph at $p_c = 1$, allowing us to vary the diameter of the graph without changing its density [11].

As shown in Figure 3 (a), as a network's diameter shrinks, FIFO's performance improves, while LIFO remains mostly unchanged. FIFO's propagation improvement is due to the decrease in path lengths. As the probability that a message is going to be sent along a long path decreases quickly, the shorter path lengths are beneficial. LIFO sorting is generally not affected by the length of the sending path because of synchronization. Most facts are not sent along long paths and a few facts are duplicated so heavily that they will almost certainly be sent to the whole network. The effect is reduced significantly for streaming for both LIFO and FIFO.

Next, we want to test the impact of network density on sorting performance. We run simulations on density-varying small world networks using 5000 factoids in the system. Small world graphs are not only parameterized by a rewiring probability, but a starting number of neighbors for each node. This



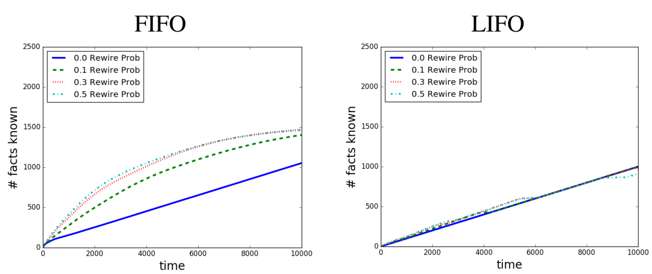
(a) Burst



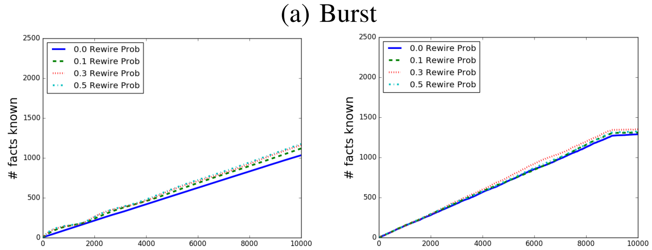
(b) Stream

Fig. 2.

This figure shows the distribution of single message branching factors for in a 10,000 step simulation as a heat map. The x-axis is the fact-id number and the y-axis is the number of copies made of that fact. The darker shades show a higher concentration of facts copied. In addition, the probability distribution of the number of copies made is shown to the right. We exclude facts copied zero times.



(a) Burst



(b) Stream

Fig. 3. Network diameter's affect on FIFO and LIFO using small world rewiring probabilities (smaller rewire probability = longer diameter)

allows us to vary the density of the graph without significantly manipulating other parts of the networks structure.

As shown in Figure 4 (a), in a high traffic scenario, as a network's density increases, the performance of both LIFO and FIFO decrease. This slow down in spreading novel information is simply due to an increase in information overload. As all actors degrees increase and their capacity to send is held stable, more information will enter their inboxes than each can handle.

B. Removing Duplicate Messages

In continuing to better our understanding of networked LIFO and FIFO sorting, we make slight additions to the basic sorting algorithms. We now consider the performance change of each sorting method when duplicate messages are removed by the underlying system. We will call these sorting methods LIFOND and FIFOND. In LIFOND, whenever a duplicate message is received, it is put on top and all other copies are removed; thus, maintaining the original last in first out order. In FIFOND, the duplicate message is just dropped, leaving the older copy in its previous location in the inbox queue.

As shown in Figure 5 and Figure 6, removing duplicate messages effectively converges the performance differences between LIFO and FIFO, making LIFO sorting perform slightly better in the burst setting and FIFO sorting perform slightly worse in the burst setting. In the stream setting, LIFO sorting performs marginally better, and FIFO sorting is unchanged. As previously discussed, LIFO sorting can cause large stacks of duplicate or already sent information to build

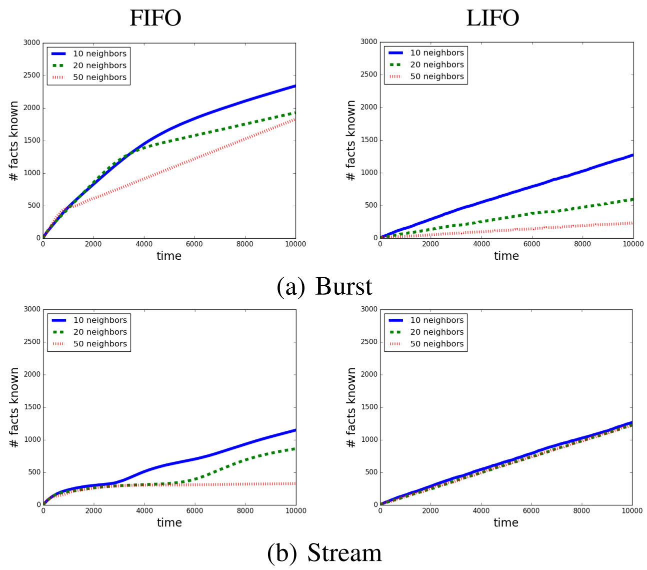


Fig. 4. Network density's affect on FIFO and LIFO using different number neighbors in small world graphs (smaller number of neighbors = lower density)

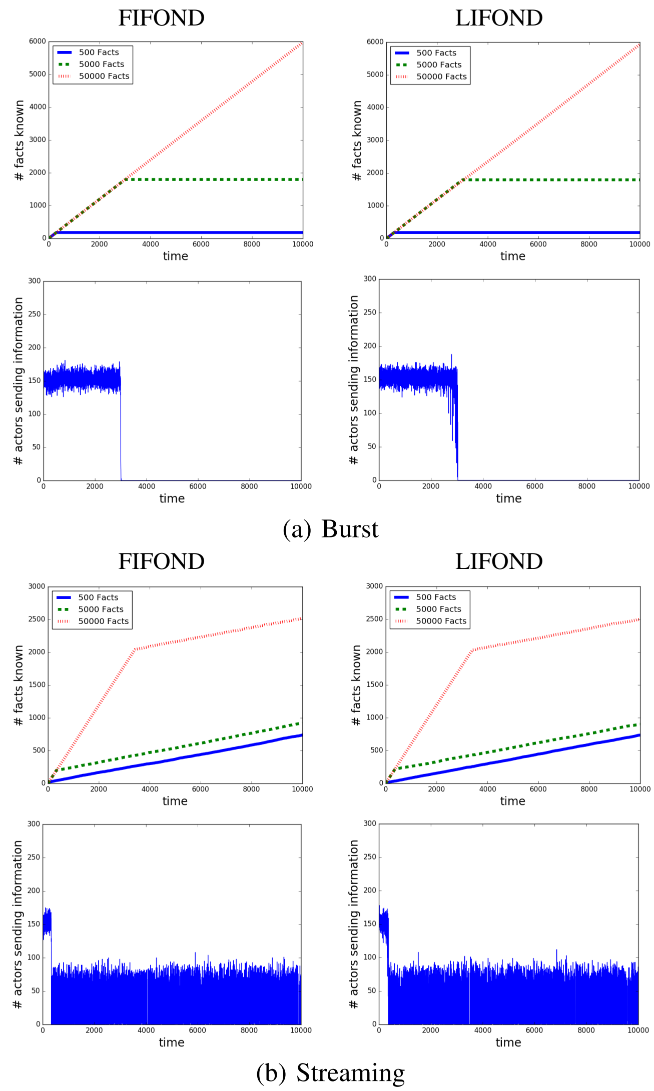


Fig. 5. Number of facts known by an actor for different amount of information in the network and number of actors sending information for 5000 factoids in the network over time for burst (top) and stream (bottom) information arrival types using FIFOND/LIFOND inbox sorting methods.

up, preventing the quick diffusion of novel information in the network. Simply suppressing duplicate information combats part of this issue. There is still a higher probability to see a message that was already sent within a short time frame, but now there are no duplicate copies stacking up. On the other hand, in the burst setting, FIFO sorting is very slightly harmed by the removal of duplicate information as there are less chances to forward a piece of information, causing more information to be filtered out quickly. Even more interestingly, the oscillatory behavior of LIFOND and FIFOND is roughly the same. This convergence further illustrates that the affect of message duplication is an important differentiation between the networked behavior of LIFO and FIFO sorting.

While these performance changes are small, they are consistent through many test.

C. Degree Distribution and Forwarding Probability

Lastly, we consider a more realistic network scenario in which 8192 actors are connected through a Kronecker graph. Kronecker graphs are generated from many iterations of Kronecker multiplication on an initial adjacency matrix with itself. Due to the nature of Kronecker multiplication, the adjacency matrix grows with every iteration. The final result of generating a Kronecker graph displays many real world network properties, such as, heavy tail degree distributions and small diameters [14]. In this new setting, information is streamed to actors at 36 messages per 5 time units, with 80,000 total messages over 10,000 steps. This can be thought of as a relatively normal traffic time. When information enters the system, facts are distributed to actors using a power-law distribution. Along with this, actors forwarding probabilities are uniformly distributed between 0.2 and 0.6 based on the retweet rate of Twitter users in a dataset found in [13]. We run these simulations for several trials and take the average.

As shown in Figure 7, the knowledge growth and oscillation patterns (oscillation patterns not explicitly shown) of both LIFO and FIFO are similar to that of the stream arrival tests found in Figure 1b. From previous experiments, we know that the simulation time it takes for networks of similar diameter to reach the same knowledge saturation point is relative to the size of the network. Hence, as these larger scale simulations are ran longer, we expect the performance to continue in the same pattern as Figure 1b. These results being almost identical to the Small-World network tests previously discussed is surprising, as it suggests that both vastly different degree distributions and network sizes have negligible impact on the behavior of networked inbox sorting. Furthermore, this

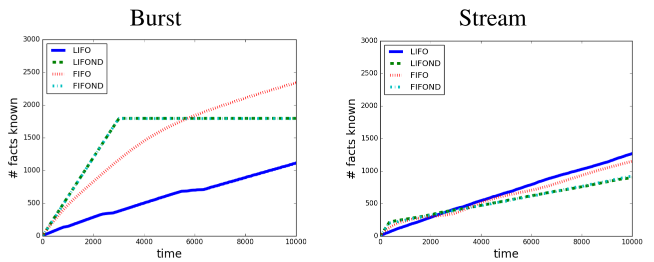


Fig. 6. Number of facts known by an actor for each sorting with 5000 factoids in the network. The Burst arrival pattern is shown on the left, the Stream arrival pattern is shown on the right. Notice that the LIFOND and FIFOND lines are the same in both graphs.

setup being more like a real-life social network in structure and forwarding probabilities provides some robustness to the other results shown in this paper.

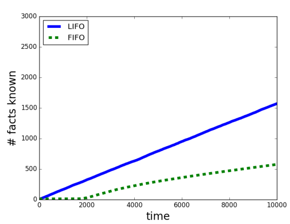


Fig. 7.
Number of facts known by an actor for LIFO and FIFO on a 8192 node Kronecker graph with stream information arrival and Twitter data forwarding.

V. Conclusions and Future Work

In conclusion, inbox sorting is a crucial factor in the dissemination of information. Our findings show a complex picture in which basic sorting algorithms in a network behave very differently for different reasons. When using a classic LIFO stack, agents are prone to becoming *synchronized* on duplicate or already seen information. This becomes a problem in bursty information arrival patterns, but seems to work well for streaming information. When using a FIFO queue, agents can gain significantly more diverse information out of the box, especially in high traffic or bursty information arrival scenarios. We find that FIFO is affected by the diameter of network structure much more than LIFO is. While we do not show here explicitly, our results remain the same for a large number of network types such as Kronecker and Barabasi-Albert graphs. In LIFO, many more copies of information are generated than strictly necessary while in FIFO likely too few are generated in information starved situations. Hence, the ideal method will depend on the information generation and consumption patterns of the users. Both of these classic sorting mechanisms do not take into account a users preferences, which is common of today's social networks. Our work illustrates that system level manipulations can help or harm access of individuals to information. We aim to explore much more intricate sorting mechanisms in future work, taking into account a user's preferences and the strength of ties. We will continue to study impact of factors such as influencers and homophily, and explore curation algorithms in feed-based networks with the goal of improving the quality of information in the network. While this study has some limitations, we hope it can serve as a building block to diversify information in networks using system level manipulations.

Acknowledgment. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] L. A. Dabbish and R. E. Kraut, "Email overload at work: An analysis of factors associated with email strain," in *CSCW*, pp. 431–440, 2006.
[2] B. Goncalves, N. Perra, and A. Vespignani, "I disrespectfully agree: The differential effects of partisan sorting on social and issue polarization," *American J. of Political Sci.*, vol. 59, pp. 128–145, 2015.
[3] M. Zomorodi, ""so your facebook friend said something racist. again."," 2015.
[4] M. Gomez-Rodriguez, K. Gummadi, and B. Schölkopf, "Quantifying information overload in social media and its impact on social contagions," in *ICWSM*, 2014.
[5] E. Bakshy, S. Messing, and L. Adamic, "Exposure to ideologically diverse news and opinion on facebook," *Science*, vol. 384, no. 6239, pp. 1130–1132, 2015.
[6] Z. Tufekci, "Algorithms in our midst: Information, power and choice when software is everywhere," in *CSCW*, pp. 1918–1918, ACM, 2015.
[7] M. Eslami, A. Aleyasen, K. Karahalios, K. Hamilton, and C. Sandvig, "Feedvis: A path for exploring news feed curation algorithms," in *ACM Conf. Companion on Computer Supported Cooperative Work & Social Computing*, pp. 65–68, ACM, 2015.
[8] P. M. Gade and C.-K. Hu, "Synchronous chaos in coupled map lattices with small-world interactions," *Physical Review E*, vol. 62, no. 5, p. 6409, 2000.
[9] L. F. Lago-Fernández, R. Huerta, F. Corbacho, and J. A. Sigüenza, "Fast response and temporal coherent oscillations in small-world networks," *Physical Review Letters*, vol. 84, no. 12, p. 2758, 2000.
[10] M. Kuperman and G. Abramson, "Small world effect in an epidemiological model," *Physical Review Letters*, vol. 86, pp. 2909–2912, 2001.
[11] D. Watts, "Collective dynamics of small-world networks," *Letters to Nature*, vol. 393, pp. 440–442, 1998.
[12] C. Moore and M.E. Newman, "Epidemics and percolation in small-world networks," *Physical Review E*, vol. 61, no. 5, p. 5678, 2000.
[13] B.D. Horne, D. Nevo, J. Freitas, H. Ji, and S. Adali, "Expertise in social networks: how do experts differ from other users?" *The International Conference on Web and Social Media*, 2016.
[14] J. Leskovec, D. Chakrabarti, J. Klienberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks" *Journal of Machine Learning Research*, vol. 11, pp. 9851042, 2010.
[15] S. Sreenivasan, K. Chan, A. Swami, G. Kornis, and B. Szymanski, "Information cascades in feed-based networks of users with limited attention" *arXiv preprint arXiv:1508.06950*, 2015.
[16] K. Chan and S. Adali, An agent based model for trust and information sharing in networked systems, *Proceedings of IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2012.