

Tight Bounds for Algebraic Gossip on Graphs

Michael Borokhovich Chen Avin Zvi Lotker

Abstract—We study the stopping times of gossip algorithms for network coding. We analyze algebraic gossip (i.e., random linear coding) and consider three gossip algorithms for information spreading: Pull, Push, and Exchange. The stopping time of algebraic gossip is known to be linear for the complete graph, but the question of determining a tight upper bound or lower bounds for general graphs is still open. We take a major step in solving this question, and prove that algebraic gossip on any graph of size n is $O(\Delta n)$ where Δ is the maximum degree of the graph. This leads to a tight bound of $\Theta(n)$ for bounded degree graphs and an upper bound of $O(n^2)$ for general graphs. We show that the latter bound is tight by providing an example of a graph with a stopping time of $\Omega(n^2)$. Our proofs use a novel method that relies on Jackson’s queuing theorem to analyze the stopping time of network coding; this technique is likely to become useful for future research.

I. INTRODUCTION

Randomized gossip-based protocols are attractive due to their locality, simplicity, and structure-free nature, and have been offered in the literature for various tasks, such as ensuring database consistency and computing aggregate information [12], [13], [2]. Consider the case of a connected network with n nodes, each holding a value it would like to share with the rest of the network. Motivated by wireless networks and limited resource sensor nodes, in recent years researchers have studied the use of randomized gossip algorithms together with network coding for this multicast task [16], [14]. The use of network coding protocols for multicast has received growing attention due to the ability of such protocols to significantly increase network capacity. For a basic network coding example, see [7].

In this work we consider *algebraic gossip*, a gossip-based network coding protocol known as random linear coding [10]. In the discussion on gossip-based protocols we distinguish between the *gossip algorithm* and the *gossip protocol*. A gossip algorithm is a communication scheme in which at every timeslot, a random node chooses a random neighbor to communicate with. We consider three known gossip algorithms: PUSH: a message is sent *to* the neighbor, PULL: a message is sent *from* the chosen neighbor, and EXCHANGE: the two nodes exchange messages. The gossip protocol, on the other hand, determines the *content* of messages sent. In algebraic gossip protocol, the content of the messages is a random linear combination of all messages stored by the sending node. Once a node has received enough independent messages (independent linear equations) it can solve the system of linear equations and discover all the initial values of all other nodes.

We study the performance of algebraic gossip on arbitrary network topologies, where information is disseminated from

all nodes in the network to all nodes, i.e., all-to-all dissemination. Previously, algebraic gossip was considered with PUSH and PULL gossip algorithms, here we also study the use of EXCHANGE, which can lead to significant improvements for certain topologies (as we show). Our main goal is to find tight bounds for the stopping time of the algebraic gossip protocol, both in expectation and with high probability (w.h.p.), i.e., with probability of at least $1 - \frac{1}{n}$.

The stopping time question, i.e., bounding the number of rounds until protocol completeness, has been addressed in the past. Deb *et al.* [6] studied algebraic gossip using PULL and PUSH on the complete graph and showed a tight bound of $\Theta(n)$, a linear stopping time, both in expectation and with high probability. Boyd *et al.* [2], [3] studied the stopping time of a gossip protocol for the *averaging problem* using the EXCHANGE algorithm. They gave a bound for symmetric networks that is based on the second largest eigenvalue of the transition matrix or, equally, the mixing time of a random walk on the network, and showed that the mixing time captures the behavior of the protocol. Mosk-Aoyama and Shah [18] used a similar approach to [2], [3] to analyze algebraic gossip on arbitrary networks. They consider symmetric stochastic matrices that (may) lead to a non-uniform gossip and gave an upper bound for the PULL algorithm that is based on a measure of conductance of the network. As the authors mentioned, the offered bound is not tight, which indicates that the conductance-based measure does not capture the behavior of the protocol.

A recent independent work by Vasudevan and Kulkarni [20] also offered the use of EXCHANGE together with algebraic gossip. Moreover, the authors give a uniform strong bound on algebraic gossip for arbitrary networks: $O(n \log n)$ in expectation and $O(n \log^2 n)$ with high probability. The worst case performance of algebraic gossip was not previously addressed in the literature.

Overview of Our Results

The main contribution of this paper is new tight bounds for the stopping time of algebraic gossip on arbitrary graphs¹. First, in Theorem 4 we disprove the results of [20] by providing a graph for which algebraic gossip takes $\Omega(n^2)$ rounds. Our main result then, Theorem 1, gives an upper bound of $O(\Delta n)$ for the stopping time of algebraic gossip on any graph, where Δ is the maximum degree in the graph.

Theorem 1. *For the asynchronous (synchronous) time model and for any graph G of size n with maximum degree Δ , the stopping time of algebraic gossip is $O(\Delta n)$ rounds both in expectation and with high probability.*

Authors are with the Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Israel, e-mails: {borokhom, avin, zvilo}@cse.bgu.ac.il.

¹The short version of this paper appeared in the Proceedings ISIT 2010 [1].

This result immediately leads to two interesting corollaries. In Corollary 1 we state a matching upper bound to Theorem 4 : for any graph of size n , since the max degree $\Delta = n$, algebraic gossip will stop w.h.p. in $O(n^2)$ rounds. In Corollary 2 we conclude a strong tight bound for *any* constant degree network (i.e., Δ is constant) of $\Theta(n)$. This improves upon known previous upper bounds which, for certain constant degree graphs, had an upper bound of $O(n^2)$. Note that the bound of $O(\Delta n)$ is not tight for all graphs (e.g., the complete graph) and the question of determine the properties of a network that capture tightly the stopping time of algebraic gossip is still open.

The second contribution of the paper is the technique we use to prove our results. We novelly bound the stopping time of **algebraic gossip** via reduction to network of queues and by the *stationary* state of the network that follows from *Jackson's theorem* for an open network of queues. The idea of using a queuing theory approach for network coding analysis was first introduced in [15] but, as opposed to our approach, it did not include the aspect of gossip communication model. We believe that the type of reduction presented in this work could be used for future analysis of gossip protocols.

Third, we compare three gossip algorithms: PUSH,PULL, and EXCHANGE. While traditionally algebraic gossip used PULL or PUSH as its gossip algorithms, it was unclear if using EXCHANGE can lead to significant improvements. We give a surprising affirmative answer to this question and prove that using the EXCHANGE algorithm can be unboundedly better than using PULL or PUSH. We show that while the time it takes to the EXCHANGE algorithm to complete the algebraic gossip on the Star graph is linear i.e. $\Theta(n)$ the time it takes the PULL and PUSH algorithms to finish the same task is $\Omega(n \log n)$. On contrary, there are many other graphs such as the Complete graph, and all the bounded degree graphs (see Section IV) that these three gossip algorithms have the same asymptotical behavior.

The rest of the paper is organized as follows. In Section II we present the communication and time models, define gossip algorithms, gossip protocols, and formally state the *gossip stopping problem*. In Section III we show that algebraic gossip on the Ring graph is linear using Jackson's theorem. In Section IV we prove our main results: a tight upper bound for arbitrary networks and a tight linear bound for graphs with a constant maximum degree. Section V gives an answer to the question: 'Is EXCHANGE can be better than PUSH or PULL?' by providing a topology for which EXCHANGE is unboundedly faster. We conclude in Section VI.

II. PRELIMINARIES AND MODELS

A. Network and Time Model

We model the communication network by a connected undirected graph $G = G(V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges. Let $N(v) \subseteq V$ be a set of neighbors of node v and $d_v = |N(v)|$ its degree, let $\Delta = \max_v d_v$ be the maximum degree of G .

The time is assumed to be slotted. We consider n consecutive timeslots as one *round*. We consider the following time models:

- **Asynchronous time model.** At every timeslot, a node selected independently and uniformly at random takes an action and a single pair of nodes communicates². In this model there is no guarantee that a node will be selected exactly once in a round; nodes can be selected several times or not at all.
- **Synchronous time model.** At every round, a random permutation of the nodes is selected uniformly and timeslots in the round are taken by nodes according to their order in the permutation. At each timeslot, a single pair of nodes communicates. The model guarantees that during n consecutive timeslots each node will be selected exactly once.

B. Gossip Algorithms

A *gossip algorithm* defines the way information is exchanged or spread in the network. At each timeslot a single node takes an information spreading action that is divided into two phases: (i) choosing a communication partner and (ii) spreading the information. A *communication partner* $u \in N(v)$ is chosen by node $v \in V$ with probability p_{vu} . Throughout this paper we will assume *uniform gossip algorithms*, i.e., $p_{vu} = \frac{1}{d_v}$.

We distinguish three gossip algorithms for information spreading between v and u , PUSH,PULL, and EXCHANGE as explained in the Introduction. We assume that messages sent in timeslot t are received in timeslot t and can be forward or processed at timeslot $t + 1$.

C. Algebraic Gossip

A *gossip protocol* is a task that is being executed using gossip algorithms, for example, calculation aggregate functions, resource discovery, and database consistency. We now describe the algebraic gossip protocol for the multicast task: disseminating n initial values of the nodes to all n nodes.

Let \mathbb{F}_q be a field of size q , each node $v_i \in V$ holds an initial value x_i that is represented as a vector in \mathbb{F}_q^r . We can represent every message as an integer value bounded by M , and therefore, $r = \lceil \log_q(M) \rceil$. All transmitted messages have a fixed length and represent linear equations over \mathbb{F}_q . The variables of these equations are the initial values $x_i \in \mathbb{F}_q^r$ and a message contains the coefficients of the variables and the result of the equation, therefore the length of each message is: $r \log_2 q + n \log_2 q$ bits. A message is built as a random linear combination of all messages stored by the node and the coefficients are drawn uniformly at random from \mathbb{F}_q . A received message will be appended to the node's stored messages only if it is independent of all linear equations (messages) that are already stored by the node and otherwise it is ignored. Initially, node v_i has only one linear equation that consists of only one variable corresponding to x_i multiplied by a coefficient 1 and equal to the value of x_i , i.e., the node knows only its initial value. Once a node receives n independent

²Alternately, this model can be seen as each node has a clock which ticks at the times of a rate 1 Poisson process and there is a total n clock ticks per round [2].

equations it is able to decode all the initial values and thus completes the task.

For a node v at timeslot t , let $S_v(t)$ be the subspace spanned by the linear equations (or vectors) it stores (i.e., the coordinates of each vector are the coefficients of the equation) at the beginning of timeslot t . The dimension (or rank) of a node is the dimension of its subspace, i.e., $\dim(S_v(t))$ and it is equal to the number of independent linear equations stored by the node.

We say that a node v is a **helpful node** to node u at the timeslot t if and only if $S_v(t) \not\subset S_u(t)$, i.e., iff a random linear combination constructed by v can be linearly independent with all equations (messages) stored by u . We call a message a **helpful message** if it increases the dimension of the node. The following lemma, which is a part of Lemma 2.1 in [6], gives a lower bound for the probability of a message sent by a **helpful node** to be a **helpful message**.

Lemma 1 ([6]). *Suppose that node v is helpful to node u at the beginning of the timeslot t . If v transmits a message to u at the timeslot t , then:*

$$\Pr(\dim(S_u(t+1)) > \dim(S_u(t))) \geq 1 - \frac{1}{q}.$$

that is, the probability of the message to be helpful is at least $1 - \frac{1}{q}$.

D. The Gossip Stopping Problem

Our goal is to compute bounds on time and number of messages needed to be sent in the network to complete various gossip protocols over various gossip algorithms. For this purpose we define the following:

Definition 1 (Excepted and high probability stopping times). *Given a graph G , gossip algorithm \mathcal{A} , and a gossip protocol \mathcal{P} , the stopping time $T(\mathcal{A}, \mathcal{P}, G)$ is a random variable defined as the number of timeslots by which all nodes complete the task. $E[T(\mathcal{A}, \mathcal{P}, G)]$ is the expected stopping time and the high probability stopping time \hat{T} is defined as follows:*

$$\hat{T}(\mathcal{A}, \mathcal{P}, G) = \min_{t \in \mathbb{Z}} [t \mid \Pr(T(\mathcal{A}, \mathcal{P}, G) \leq t) \geq 1 - \frac{1}{n}]$$

We can now express our research question formally:

Definition 2 (Gossip stopping problem). *Given a graph G , a gossip algorithm \mathcal{A} , and a gossip protocol \mathcal{P} , the gossip stopping problem is to determine $E[T(\mathcal{A}, \mathcal{P}, G)]$ and $\hat{T}(\mathcal{A}, \mathcal{P}, G)$, the expected and high probability stopping times.*

In this work we consider $\mathcal{A} \in \{\text{PUSH}, \text{PULL}, \text{EXCHANGE}\}$ and $\mathcal{P} = \text{algebraic gossip}$, so when these parameters and G are understood from the context, we denote the expected and high probability stopping times as $E[T]$ and \hat{T} respectively. Moreover, we usually measure the stopping time in *rounds* where one round equals n consecutive timeslots. Thus, we define the expected number of rounds as $E[R] = E[T]/n$ and $\hat{R} = \hat{T}/n$ as the number of rounds by which all nodes complete the task with high probability.

For clarity, we present our proofs for the *asynchronous* time model and the EXCHANGE algorithm, we extend the results to the *synchronous* cases and PUSH and PULL in the appendix where we also put some of the more technical proofs.

III. LINEAR BOUND ON A RING VIA QUEUING THEORY

Before proving the main results of Theorem 1 in the next section we prove in this section a bound on the specific case of a Ring network. This is a simpler case to prove and understand, and will be used as a basis for the proof of the general result. A Ring of size n is a connected cycle where each node has a one left and a one right neighbor.

Theorem 2. *For the asynchronous time model and the Ring graph of size n , the stopping time (measured in rounds) of algebraic gossip is linear with high probability, and in expectation i.e., $\hat{R} = \Theta(n)$ and $E[R] = \Theta(n)$.*

Proof: The idea of the proof is to reduce the problem of network coding on the Ring graph to a simple system of queues and use Jackson's theorem for open networks to bound the time it takes *helpful messages* to cross the network.

To simplify our analysis, we cut the Ring in an arbitrary place and get a Path graph (without loss of generality, we assume that the leftmost node in the Path is v_1 and the rightmost node is v_n), see Fig. 1 (a). It is clear that the stopping time of the algebraic gossip protocol will be larger in a Path graph than in a Ring graph. Another simplification that we will do, for the first part of the proof, is to consider only the messages that travel from left to right (towards v_n) (i.e., other messages will be ignored, thus increasing the stopping time).

We define a queuing system by assuming a queue with a single server at each node. Customers of our queuing network are the *helpful messages*, i.e., messages that increase the rank of a node they arrive at. This means that every customer arriving at some node increases its rank by 1, so the queue size at a node represents a measure of *helpfulness* of the node to its right-hand neighbor (i.e., the queue size is the number of independent linear equations that the node can generate for its right-hand neighbor). The service procedure at node v_i is a transmission of a *helpful message* (customer) from v_i to v_{i+1} . So, from Lemma 1, the probability that a customer will be serviced at node v_i in a given timeslot is: $p \geq \frac{1}{n}(1 - \frac{1}{q})$, where $\frac{2}{n} \cdot \frac{1}{2} = \frac{1}{n}$ is the probability that in the EXCHANGE algorithm a message will be sent from v_i to v_{i+1} at any given timeslot.

Thus, we can consider that a service time in our queuing system is geometrically distributed with parameter p . The service time is distributed over the set $\{0, 1, 2, \dots\}$, which means that a customer that enters an empty queue at the *end* of the timeslot can be immediately serviced with probability p (since it is the beginning of the next timeslot). A customer cannot pass more than one node (queue) in a single timeslot so, we define the transmission time as one timeslot. I.e., the time needed for a customer to pass through k queues is the sum of the waiting time in each queue, service time in each queue and additional k timeslots for transmission from queue to queue.

The following lemma shows that the service rate can be bounded from below by an exponential random variable. The proof of this lemma can be found in Appendix C.

Lemma 2. *Let X be a geometric random variable with parameter p and supported on the set $\{0, 1, 2, \dots\}$, i.e., for*

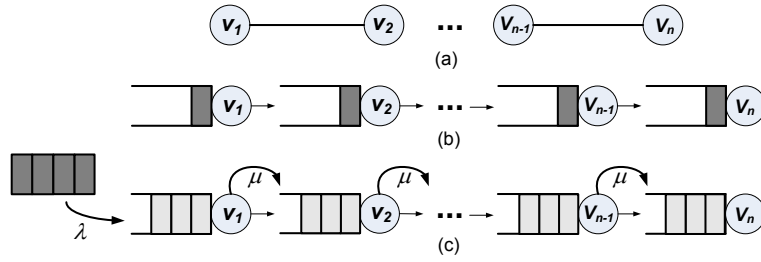


Fig. 1. Modeling algebraic gossip in a Path as a queuing network. (a) – Initial Path graph. (b) – One real customer at each node. (c) – Queues are filled with dummy customers and real customers enter the system from outside.

$k \in \mathbb{Z}^+$: $\Pr(X = k) = (1-p)^k p$, and let Y be an exponential random variable with parameter p . Then, for all $x \in \mathbb{R}^+$:

$$\Pr(X \leq x) \geq \Pr(Y \leq x) = 1 - e^{-px}, \quad (1)$$

i.e., a random variable $Y \sim \text{Exp}(p)$ stochastically dominates the random variable $X \sim \text{Geom}(p)$.

We can now assume that the service time is exponentially distributed with parameter $\mu = p$. This assumption decreases the rate of transmission of *helpful messages*, and therefore will only increase the stopping time. The last is true since the probability that a customer will be serviced by the time t_1 in a geometrical server is higher than in an exponential server, and thus, each customer in a network with geometric servers will arrive at v_n by time t_2 with higher probability than in a network with exponential servers. The formal justification of this step is given later in Lemma 5 which prove this assertion for trees and not only for the line.

To this end, we have converted our network to a standard network of queues where the network is open, external arrivals to nodes will form a Poisson process, service times are exponentially distributed and the queues are first come first serve (FCFS). For a queue i let μ_i denote the service rate and λ_i the total arrival rate. We present now Jackson's theorem for open networks; a proof of this theorem can be found in [4].

Jackson's Theorem. *In an open Jackson network of n queues where the utilization $\rho_i = \frac{\lambda_i}{\mu_i}$ is less than 1 at every queue, the equilibrium state probability distribution exists, and for state (k_1, k_2, \dots, k_n) is given by the product of the individual queue equilibrium distributions: $\pi(k_1, k_2, \dots, k_n) = \prod_{i=1}^n \rho_i^{k_i} (1 - \rho_i)$.*

We would like to use Jackson's theorem to conclude that there is an equilibrium state for our network of queues and that in the equilibrium state the lengths of the queues are independent. For Jackson's theorem to hold we need to appropriately define the arrival rate to the queues, so we will slightly change our queuing network.

The initial state of our system is that at every queue we have one *real* customer (see Fig. 1 (b)). Now we take all the n *real* customers out from the system and let them enter back via the leftmost queue with a predefined arrival rate. Clearly, this modification increases the stopping time. We define the *real* customers' arrivals as a Poisson process with rate $\lambda = \frac{\mu}{2}$. So, $\rho_i = \frac{\lambda_i}{\mu_i} = \frac{1}{2} < 1$ for all queues ($i \in [1..n]$).

Now, according to Jackson's theorem there exists an equilibrium state. So, our last step is to ensure that the lengths

of all queues at time $t = 0$ are according to the equilibrium state probability distribution. We add *dummy* customers to all the queues according to the stationary distribution. By adding additional *dummy* customers (we call them *dummy* since their arrivals are not counted as a rank increment) to the system, we make the *real* customers wait longer in the queues, thus increasing the stopping time. Our queuing network with the above modifications is illustrated in Fig. 1 (c), where the *real* customers are dark, and the *dummy* customers are bright.

We will compute the stopping time in two phases. By the end of the first phase, the node v_n will finish the algebraic gossip task. By the end of the second phase, all the nodes will finish the task. For the first phase, we will find the time it takes the n 'th (last) *real* customer to arrive at the rightmost node, i.e., node v_n . By that time, the rank of node v_n will become n and it will finish the algebraic gossip protocol (i.e., it received n *helpful messages*). Let us denote this time (in *timeslots*) as $T^a + T^b$, where T^a is the time needed for the n 'th customer to arrive at the first queue, and T^b is the time needed for the n 'th customer to pass through all the n queues in the system.

For the second phase, let us assume that after $T^a + T^b$ timeslots (when v_n finishes the algebraic gossip task) all nodes except node v_n forget all the information they have. So, the rank of all nodes except v_n is 0. Let us now analyze the information flow from the rightmost node in the Path (v_n) to the leftmost node (v_1). In the same way, we will represent all *helpful messages* that node v_n will send as customers in our queuing system. In order to use Jackson's Theorem, we will again remove all the *real* customers from the system and will inject them to the queue of node v_n with a Poisson rate $\lambda = \mu/2$. We also fill all the queues in the system with *dummy* customers in order to achieve queue lengths that correspond to the equilibrium state distribution. Clearly, arrival of a *real* customer at some node v_i ($i \neq n$) will increase the rank of that node. So, after the last *real* customer arrives at node v_1 , the ranks of *all* nodes will be n , and the algebraic gossip task will be finished.

Using the same equilibrium state analysis as before, we define the time it takes the last *real* customer to arrive at the rightmost node v_n as T^c , and the time to cross all the n queues – arriving at node v_1 – as T^d .

So, $T = T^a + T^b + T^c + T^d$, is an upper bound for the number of timeslots needed to complete the task. Now we will find upper bound for T^x , $x \in \{a, b, c, d\}$ and then we will use union bound to obtain an upper bound on T .

From Jackson's Theorem, it follows that the number of customers in each queue is independent, which implies that the random variables that represent the waiting times in each queue are independent. To continue with the proof we need the following lemmas; the first is a classical result from queuing theory, the proof of the second lemma can be found in Appendix D.

Lemma 3 ([19], section 4.3). *Time needed to cross one M/M/1 queue in the equilibrium state has an exponential distribution with parameter $\mu - \lambda$.*

Lemma 4. *Let Y be the sum of n independent and identically distributed exponential random variables (each one with parameter $\mu > 0$), and $E[Y] = \frac{n}{\mu}$. Then, for $\alpha > 1$:*

$$\Pr(Y < \alpha E[Y]) > 1 - (2e^{-\alpha/2})^n. \quad (2)$$

Recall that: $\mu = p \geq \frac{1}{n}(1 - \frac{1}{q})$ so $\mu \geq \frac{q-1}{qn} \geq \frac{1}{2n}$ for $q \geq 2$. The random variable T^a is the sum of n independent random variables distributed exponentially with parameter $\mu/2$. From Lemma 3 we obtain that T^b is the sum of n independent random variables distributed exponentially with parameter $\mu - \lambda = \frac{\mu}{2}$. It is clear that T^c is distributed exactly as T^a and T^d is distributed exactly as T^b . Therefore (for $\mu = \frac{1}{2n}$): $E[T^x] = \sum_{i=1}^n \frac{2}{\mu} = 4n^2$. Using Lemma 4 (with $\alpha = 2$) we obtain for $x \in \{a, b, c, d\}$:

$$\Pr(T^x \leq 8n^2) \geq 1 - \left(\frac{2}{e}\right)^n. \quad (3)$$

Using union bound, we get that:

$$\Pr(T \leq 32n^2) \geq \Pr(\cap_x T^x \leq 8n^2) \quad (4)$$

$$= 1 - \Pr(\cup_x T^x > 8n^2) \quad (5)$$

$$\geq 1 - 4 \left(\frac{2}{e}\right)^n \quad (6)$$

where $x \in \{a, b, c, d\}$. It is clear that $\Pr(T \leq 32n^2)$ increases when μ increases (faster server yields smaller waiting time); hence, the above inequality holds for any $\mu \geq \frac{1}{2n}$.

So, for the asynchronous time model and EXCHANGE we obtain an upper bound for the high probability stopping time: $\hat{T} = O(n^2)$ in timeslots, and thus $\hat{R} = O(n)$, in rounds. Let us now find an upper bound for the expected number of rounds needed to complete the task – $E[R]$:

$$E[R] = \frac{1}{n} E[T] \quad (7)$$

$$= \frac{1}{n} E[T^a + T^b + T^c + T^d] \quad (8)$$

$$= \frac{4}{n} E[T^x] = \frac{4}{n} 4n^2 = 16n = O(n). \quad (9)$$

The lower bound is clear since in order to finish the algebraic gossip task each node has to receive at least n messages, so at least n^2 messages need to be sent and received. Since in each timeslot at most 2 messages (using EXCHANGE) are sent, we get: $\hat{T} = \Omega(n^2)$, thus $\hat{R} = \Omega(n)$, and $E[R] = \Omega(n)$.

The result of the Theorem 2 is then follows: $E[R] = \Theta(n)$, and $\hat{R} = \Theta(n)$. ■

IV. ALGEBRAIC GOSSIP ON ARBITRARY GRAPHS

Now we are ready to prove our main results. First, we present the upper bound for any graph as a function of its

maximum degree Δ , and then we give corollaries that are applications of this result for more specific cases.

Theorem 1 (restated). *For the asynchronous time model and for any graph G of size n with maximum degree Δ , the stopping time of algebraic gossip is $O(\Delta n)$ rounds both in expectation and with high probability.*

Proof: Consider an arbitrary graph G of size n with a maximum degree Δ and a vertex v . First, let us perform a (Breadth First Search) BFS on G starting at v . The search results with a directed shortest path spanning tree G_v , rooted at v where each node has a single edge toward its parent in the tree (except the root v). In the analysis we will only consider messages that sent on the edges of the spanning tree in the direction toward v , i.e., we ignore all messages received from non tree edges or in the opposite direction (see Fig. 2 (b)).

Now, let us concentrate on the information flow towards the node v from all other nodes. As in the proof of Theorem 2, we will define a queuing system with a queue at each node (see Fig. 2 (c)). The following lemma shows that we can model the service time at each queue as an exponential random variable with parameter $\mu = p$.

Let G_v be a tree of size n rooted by node v . Let $\mathcal{N}(G_v, \mathcal{S})$ be a network of n queues where for each node u in G_v there is a queue and the queue output is connected to the input of the queue corresponding to the parent of u in G_v . In addition, each queue is of infinite size and initially has one customer in the queue (see Fig. 2) (c)). The servers of all the queues work with a service time distributed as \mathcal{S} . Let $T(G_v, \mathcal{S})$ be the random time by which all the n customers in $\mathcal{N}(G_v, \mathcal{S})$ arrive to the queue of v (we assume v does not serve the customers).

Lemma 5. *For any tree G_v and $0 < p \leq 1$:*

$$\Pr(T(G_v, \text{Geom}(p)) \leq t) \geq \Pr(T(G_v, \text{Exp}(p)) \leq t).$$

The result of this lemma is that any probabilistic upper bound on the stopping time of v in a tree network with exponential servers holds for the same tree network with geometric servers (both with the same parameter p and initially one customer at each queue).

Once all *real* customers arrive at v , it will reach rank n and will finish the algebraic gossip task. Now we have to calculate the service time parameter p . The degree of each node in G is at most Δ . Each node in G_v , except v , has a parent. Since we virtually remove (i.e., ignore) all edges that do not belong to G_v , at each node there is exactly one edge that goes towards the root v . Therefore, the probability that a customer will be serviced (transmitted towards v) at the end of a given timeslot is at least: $p \geq \left(\frac{2}{n} \cdot \frac{1}{\Delta}\right) \left(1 - \frac{1}{q}\right)$, where $\frac{2}{n} \cdot \frac{1}{\Delta} = \frac{2}{n\Delta}$ is the probability that in the EXCHANGE algorithm a message will be sent on the edge that goes towards v during one timeslot, and $\left(1 - \frac{1}{q}\right)$ is the minimal probability that the message is *helpful* (Lemma 1). Clearly, $p \geq \frac{1}{n\Delta}$ for $q \geq 2$ so we set our exponential servers to work with rate $\mu = \frac{1}{n\Delta}$.

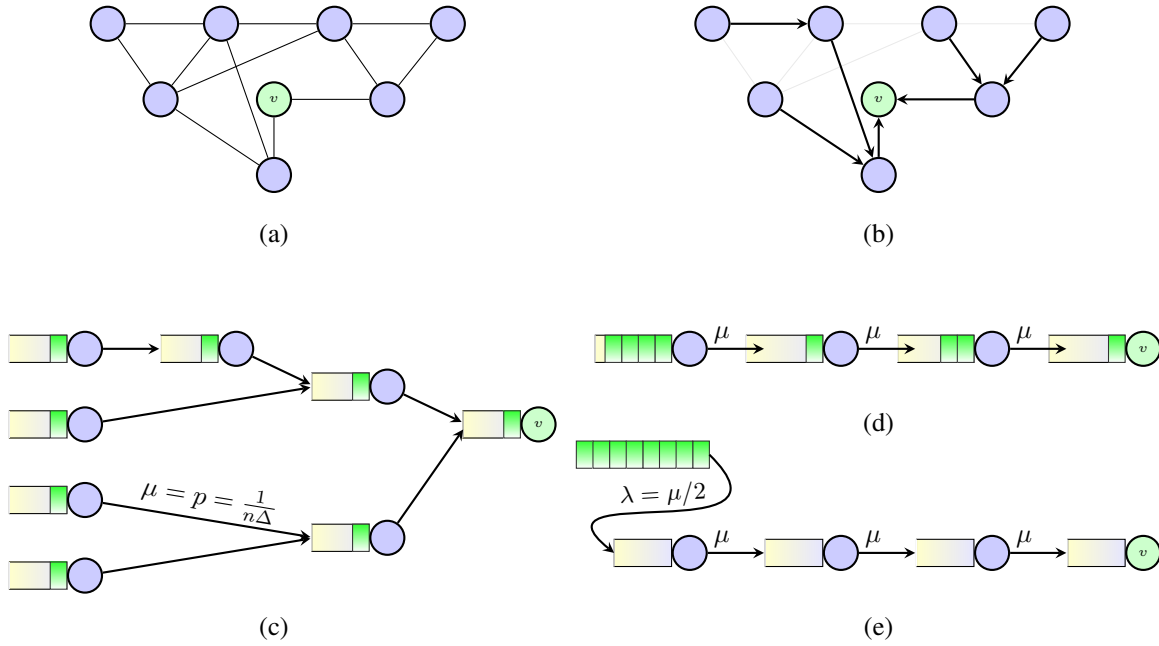


Fig. 2. Reduction of Algebraic Gossip to a system of queues. (a) – Initial graph G_n . (b) – BFS spanning tree T_n . (c) – System of queues Q_n^{tree} . (d) – System of queues $Q_{l_{\max}}^{line}$. Stopping time of $Q_{l_{\max}}^{line}$ is larger than of Q_n^{tree} . (e)–Taking all customers out of the system and use Jackson theorem for open networks.

Theorem 3. Let Q_n^{tree} be a network of n nodes arranged in a tree topology, rooted at the node v . Each node has an infinite queue, and a single exponential server with parameter μ . Initially, there is a single customer in every queue. The time by which all the n customers leave the network via the root node v is $t(Q_n^{tree}) = O(n/\mu)$ with high probability. Formally, for any $\alpha > 1$:

$$\Pr(t(Q_n^{tree}) < \alpha 4n/\mu) > 1 - 2(2e^{-\alpha/2})^n. \quad (10)$$

The main idea of the Theorem 3 proof is to show that the stopping time of the network Q_n^{tree} (i.e., the time by which all the customers leave the network) is stochastically³ smaller or equal to the stopping time of the systems of l_{\max} queues arranged in a line topology – $Q_{l_{\max}}^{line}$ (l_{\max} is the depth of the tree Q_n^{tree}). Then, we make the system $Q_{l_{\max}}^{line}$ stochastically slower by moving all the customers out and make them enter the system via the farthest queue with the rate $\lambda = \mu/2$. Finally, we use Jackson's Theorem for open networks (similar to the proof of Theorem 2) to find the stopping time of the system. See Fig. 2 for the illustration. The full proof of Theorem 3 can be found in the Appendix G.

Using Theorem 3 for the tree G_v and with $\mu = \frac{1}{\Delta n}$, we obtain the stopping time of the node v : $T_v < \alpha 4n^2\Delta$ with probability of at least $1 - 2(2e^{-\alpha/2})^n$.

The same analysis holds for any node $u \in V$, i.e., performing BFS starting from the node u , defining a queuing system on the tree G_u , and finding the stopping time of u , T_u . So, we can use a union bound to obtain the stopping time of all

the nodes in G :

$$\Pr\left(\bigcap_{u \in V} T_u < \alpha 4n^2\Delta\right) \geq 1 - 2n(2e^{-\alpha/2})^n. \quad (11)$$

By letting $\alpha = 2$ we obtain:

$$\Pr\left(\bigcap_{u \in V} T_u < 8n^2\Delta\right) \geq 1 - 2n\left(\frac{2}{e}\right)^n. \quad (12)$$

So, we obtained that the stopping time of the algebraic gossip in G is $O(\Delta n^2)$ timeslots with high probability and thus: $\hat{R} = O(\Delta n)$.

The high probability bound of Eq. (11) is true for any $\alpha > 1$ and therefore strong enough to bound the expectation (see proof in the Appendix E) and finish the proof of Theorem 1:

$$E[T] = O(\Delta n^2) \text{ and } E[R] = O(\Delta n). \quad (13)$$

From Theorem 1, and since the maximum degree is at most n we can derive a general upper bound of algebraic gossip on any graph.

Corollary 1. For the asynchronous time model and any graph G of size n , the gossip stopping time of the algebraic gossip task is $O(n^2)$ rounds with high probability and in expectation.

We can use Theorem 1 to obtain a tight linear bound of algebraic gossip on graphs with a constant maximum degree. We note that previous bounds for this case are not tight, for example, for the Ring graph the bound of [18] is $O(n^2)$.

Corollary 2. For the asynchronous time model and any bounded-degree graph G of size n with a constant maximum degree Δ , the gossip stopping time of the algebraic gossip task is linear rounds with high probability and in expectation.

³For completeness, stochastic dominance is formally defined in Appendix G1.

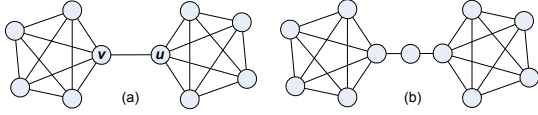


Fig. 3. (a) – Barbell graph: two cliques of size $\frac{n}{2}$ connected with a single edge. (b) – Extended barbell graph: additional node between the cliques.

We now show that the upper bound $O(\Delta n)$ presented in Theorem 1 is tight in the sense that for each $2 \leq \Delta \leq n$ there exists a graph for which algebraic gossip takes $\Omega(\Delta n)$ rounds.

Theorem 4. *For any $2 \leq \Delta \leq n/2 + 1$ and for the asynchronous time model there exists a graph G of size n with maximum degree Δ for which algebraic gossip takes $\Omega(\Delta n)$ rounds with high probability, in particular there is a graph for which the stopping time is $\Omega(n^2)$ rounds with high probability.*

Proof: We present here the proof for the case of an even n and $\Delta = n/2 + 1$ which can be easily extended to smaller Δ . In order to prove this result we will need the following lemma, the proof can be found in Appendix F:

Lemma 6. *Let X be a sum of m independent and identically distributed geometric random variables with parameter p , i.e., $X = \sum_{i=1}^m X_i$. Then, for any positive integer $k < m/p$*

$$\Pr(X > k) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}} kp} \right)^{-m}. \quad (14)$$

Let n be even and consider the Barbell graph which is build from two disjoint cliques of size $n/2$ connected by a single edge between u and v (see Figure 3 (a)). We will concentrate on the information flow from u to v . It is clear that to achieve a rank n node u has to receive $\frac{n}{2}$ helpful messages from v . Using the fastest gossip variation - EXCHANGE the probability p that a helpful message will be sent in one timeslot from v to u is: $\frac{2}{n} \cdot \frac{1}{n/2} \geq p \geq \frac{2}{n} \cdot \frac{1}{n/2} (1 - \frac{1}{q})$. Hence: $p \leq \frac{4}{n^2}$. The number of timeslots, T , needed to the node v to send $n/2$ helpful messages can be viewed as a sum of $n/2$ geometric random variables with parameter p . Clearly, $E[T] = \frac{n}{2} \cdot \frac{1}{p} \geq \frac{n^3}{8}$ timeslots. Using Lemma 6 with $k = \lfloor n^3/16 \rfloor$, $p = \frac{4}{n^2} \cdot \frac{n^3/16}{\lfloor n^3/16 \rfloor} \geq \frac{4}{n^2}$ (we took p even larger than its maximum value; this will make calculations nicer and will not affect the bound), and $m = n/2$ we get:

$$\Pr(T > \lfloor n^3/16 \rfloor) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}} kp} \right)^{-m} \quad (15)$$

$$= 1 - (\sqrt{e}/2)^n. \quad (16)$$

It is clear that $\Pr(T \geq k)$ increases when p decreases (the smaller probability of success – the larger the probability to finish later). Hence, the above inequality holds for any $p \leq 4/n^2$.

Thus, the number of timeslots needed is at least $\lfloor n^3/16 \rfloor$ w.h.p. and $n^3/8$ in expectation. So, the total stopping time of the algebraic gossip protocol over the Barbell graph (measured in rounds) is: $\hat{R} = \Omega(n^2)$, and $E[R] = \Omega(n^2)$. ■

V. EXCHANGE CAN BE UNBOUNDEDLY FASTER THAN PUSH OR PULL

As we presented earlier, there are three gossip variations: PUSH, PULL, and EXCHANGE. In PUSH or PULL there is only one message sent between the communication partners, in EXCHANGE there are two messages sent. Thus, the total message complexity for the same number of communication rounds is doubled. So, we would like to know: is the stopping time decrease, when using EXCHANGE, worth the doubling message complexity? In this section we give the answer by presenting a graph for which the EXCHANGE gossip algorithm is unboundedly faster than the PUSH or PULL algorithms.

Theorem 5. *For the Star graph S_n (which is a tree of n nodes with one node having degree $n-1$ and the other $n-1$ nodes having degree 1), algebraic gossip using EXCHANGE is unboundedly better than using PUSH or PULL algorithms. Formally, for $\mathcal{A} \in \{\text{PUSH}, \text{PULL}\}$:*

$$\lim_{n \rightarrow \infty} \frac{\hat{R}(\mathcal{A})}{\hat{R}(\text{EXCHANGE})} \rightarrow \infty, \text{ and} \quad (17)$$

$$\lim_{n \rightarrow \infty} \frac{E[R(\mathcal{A})]}{E[R(\text{EXCHANGE})]} \rightarrow \infty. \quad (18)$$

The proof of this theorem is a direct consequence of the following lemmas. The proofs of these lemmas are omitted.

Lemma 7. *For the Star graph S_n , algebraic gossip using PUSH takes $\Omega(n^2)$ rounds high probability and in expectation.*

Lemma 8. *For the Star graph S_n , algebraic gossip using PULL takes $\Omega(n \log n)$ rounds with high probability and in expectation.*

Lemma 9. *For the Star graph S_n , algebraic gossip using EXCHANGE takes $O(n)$ rounds with high probability and in expectation.*

VI. CONCLUSIONS

In this work we prove tight bounds on the stopping time of the algebraic gossip protocol. We prove that the upper bound for any graph is $O(n^2)$ and we show that this bound is tight, i.e., there exists a graph for which the stopping time of algebraic gossip is $\Omega(n^2)$. Our general upper bound is provided as a function of the maximum degree Δ of a graph and thus we can obtain a tight linear bound of $\Theta(n)$ for any graph with a constant maximum degree. Moreover our results hold for $q \geq 2$ (coefficients field size) while previous results were for the case $q \geq n$.

It is still an open question to determine the properties of a network that capture the stopping time of algebraic gossip. To illustrate this, note the interesting observation that on the extended-Barbell graph (Fig. 3 (b)) the stopping time of algebraic gossip is linear. So, by adding a single node to the graph of Fig. 3 (a) the stopping time has been changed by an order of magnitude?!

A recent work of Haeupler [9] (to appear in STOC 2011), that followed the conference version of this paper [1], answers the question we ask here. Haeupler's paper makes a significant progress in analyzing the stopping time of algebraic gossip.

While all previous works on algebraic gossip used the notion of *helpful message/node* to look at the rank evaluation of the matrices each node maintains (this approach was initially proposed by [6]), Haeupler used a completely different approach. Instead of looking on the growth of the node's subspace (spanned by the linear equations it has), he proposed to look at the orthogonal complement of the subspace and then analyze the process of its disappearing. This elegant and powerful approach led to a very impressive result: a tight bound of $\Theta(n/\gamma)$ was proposed for all-to-all communication, where γ is a min-cut measure of the related graph. This matches our tight bounds for many topologies (e.g., constant maximum degree graphs, barbell graph, etc.) and extends them for other graphs. Haeupler proposed also results for many-to-all (k to n) communication, but these bounds are not always tight. In the future work we aim to find tight bounds for the many-to-all communication with uniform algebraic gossip and also to analyze the non-uniform gossip approaches for information dissemination.

REFERENCES

- [1] M. Borokhovich, C. Avin, and Z. Lotker. Tight bounds for algebraic gossip on graphs. In *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1758–1762, jun. 2010.
- [2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006.
- [3] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. In *INFOCOM*, pages 1653–1664, 2005.
- [4] H. Chen and D. Yao. *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, volume 46 of *Applications of Mathematics*. Springer-Verlag, New York, first edition, 2001.
- [5] H. Chen and D. Yao. *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, volume 46 of *Applications of Mathematics*. Springer-Verlag, New York, first edition, 2001.
- [6] S. Deb, M. Médard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*, 52(6):2486–2507, 2006.
- [7] C. Fragouli, J.-Y. L. Boudec, and J. Widmer. Network coding: an instant primer. *Computer Communication Review*, 36(1):63–68, 2006.
- [8] G. R. Grimmett and D. R. Stirzaker. *Probability and random processes*. Oxford University Press, New York, third edition, 2001.
- [9] B. Haeupler. Analyzing Network Coding Gossip Made Easy. To appear in the *43rd ACM Symposium on Theory of Computing (STOC)*, 2011.
- [10] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *ISIT*, pages 442+, 2003.
- [11] R. V. D. Hofstad. Random graphs and complex networks. 2010.
- [12] R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *FOCS*, pages 565–574, 2000.
- [13] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *FOCS*, pages 482–491, 2003.
- [14] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.
- [15] D. S. Lun, M. Medard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. *Physical Communication*, 1(1):3–20, 2008.
- [16] M. Médard and R. Koetter. Beyond routing: An algebraic approach to network coding. In *INFOCOM*, pages 122–130, 2002.
- [17] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.
- [18] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *ISIT*, pages 1748–1752, 2006.
- [19] C.-H. Ng and S. Boon-Hee. *Queueing Modelling Fundamentals: With Applications in Communication Networks*. Wiley Publishing, 2008.
- [20] D. Vasudevan and S. Kudekar. Algebraic gossip on arbitrary networks, 2009.

APPENDIX

A. Algebraic gossip with synchronous time model, with PUSH and PULL

In this section of Appendix we give theorems and corollaries that extend the results presented in the paper to both time models (synchronous and asynchronous), and to the three gossip algorithms (PUSH, pull, and EXCHANGE).

The first theorem shows that the general upper bound for algebraic gossip holds also for the synchronous time model.

Theorem 6. *For the synchronous time model and for any graph G_n with maximum degree Δ , the stopping time of algebraic gossip is $O(\Delta n)$ rounds with high probability.*

Proof: The proof for the synchronous time model is almost the same as the asynchronous case. The analysis will be done in *rounds* instead of *timeslots*. The probability that a customer will be serviced (transmitted towards v) at the end of a given *round* is at least: $p \geq (1 - (\frac{\Delta-1}{\Delta})^2)(1 - \frac{1}{q})$, where $(1 - (\frac{\Delta-1}{\Delta})^2) = \frac{2}{\Delta} - \frac{1}{\Delta^2}$ is the probability that in the EXCHANGE algorithm at least one message will be sent on a specific edge (in a specific direction) during one *round*, and $(1 - \frac{1}{q})$ is the minimal probability that the message is *helpful* (Lemma 1).

$$p \geq (\frac{2}{\Delta} - \frac{1}{\Delta^2})(1 - \frac{1}{q}) \geq \frac{1}{\Delta}(1 - \frac{1}{q}) \geq \frac{1}{2\Delta} \text{ for } q \geq 2.$$

If the node i has received a message during a specific round from the node j it will ignore the additional message that can arrive from the same node j at the same round (this can happen if i chooses j and j chooses i in the EXCHANGE gossip scheme in one round). Clearly, this assumption can only increase the stopping time since we ignore (possibly helpful) information.

T^x , $x \in \{a, b\}$ are measured now in *rounds* and not in *timeslots*. Since $\mu = p \geq \frac{1}{2\Delta}$, using Lemma 4 (with $\alpha = 2$ and $E[T^x] = \frac{2n}{\mu} = 4n\Delta$), we obtain:

$$\Pr(T^x < 8n\Delta) > 1 - (\frac{2}{e})^n, \text{ for } x \in \{a, b\}.$$

The rest of the proof is the same as in the asynchronous case and thus the result follows. ■

The following theorem proves that the worst-case lower bound for algebraic gossip holds also for the synchronous time model.

Theorem 7. *For the synchronous time model there exists a graph G_n for which algebraic gossip takes $\Omega(n^2)$ rounds with high probability.*

Proof: The proof is almost the same as in Theorem 4. The analysis will be done in *rounds* instead of *timeslots*.

Using the fastest gossip variation - EXCHANGE the probability p that a *helpful message* will be sent in one round from v to u is: $(1 - (\frac{n/2-1}{n/2})^2) \geq p \geq (1 - (\frac{n/2-1}{n/2})^2)(1 - \frac{1}{q})$. Hence: $p \leq \frac{2}{n/2} - \frac{1}{(n/2)^2} \leq \frac{4}{n}$. Clearly, $E[R] = \frac{n}{2} \cdot \frac{1}{p} \geq \frac{n^2}{8}$ rounds.

Using Lemma 6 (with $k = \lfloor n^2/16 \rfloor$, $p = \frac{4}{n} \frac{n^2/16}{\lfloor n^2/16 \rfloor}$, and $m = n/2$), and R measured in **rounds**, we get:

$$\Pr(R > \lfloor n^2/16 \rfloor) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}kp}} \right)^{-m} = 1 - (\sqrt{e}/2)^n.$$

The rest of the proof is the same as in Theorem 4. ■

The following corollary shows that our bounds (upper and lower) for algebraic gossip on general graphs hold also for the PUSH and PULL gossip algorithms.

Corollary 3. *The results of Theorems 1, 4, 6, and 7 hold also for PUSH and PULL gossip algorithms.*

Proof: By moving from the EXCHANGE to the PUSH or PULL gossip algorithms, we change only the probability of sending a *helpful message* on a specific (directed) edge, i.e., the service time at each node will change. Easy to see that this probability will be decreased by a factor of 2 (i.e., the service time will become twice longer). Clearly, such a reduction will not affect the asymptotical bounds that were achieved using Lemmas 4, and 6. ■

B. Proof of Lemma 5

Lemma 5 (restated): *For any tree G_v and $0 < p \leq 1$:*

$$\Pr(T(G_v, \text{Geom}(p)) \leq t) \geq \Pr(T(G_v, \text{Exp}(p)) \leq t).$$

Proof: We will prove this claim by showing that for each customer c and for each queue u on the unique path that c traverse to the root the probability that c reaches u before time t is larger in $\mathcal{N}(G_v, \text{Geom}(p))$ than in $\mathcal{N}(G_v, \text{Exp}(p))$.

Consider a reverse topological order of the nodes in G_v , $v^1, v^2, \dots, v^n = v$, i.e., for every node v^i , $1 \leq i < n$, the parent of v^i is a node v^j and $j > i$. For a node v^i let C^i be the set of customers that it need to serve on their way to the root. For a node v^i and a customer $c \in C^i$ let $\mathcal{G}_c^i(t)$ denote the event that c reached v^i before time t in $\mathcal{N}(G_v, \text{Geom}(p))$ and let $\mathcal{E}_c^i(t)$ be defined similarly for $\mathcal{N}(G_v, \text{Exp}(p))$. We claim that for each $1 \leq i \leq n$, and each $c \in C^i$, $\Pr(\mathcal{G}_c^i(t)) \geq \Pr(\mathcal{E}_c^i(t))$ and the proof will be by induction on i .

Induction basis: $\Pr(\mathcal{G}_{v_1}^1(t)) \geq \Pr(\mathcal{E}_{v_1}^1(t))$. By definition v_1 is a leaf with one customer, itself, and no children so $\Pr(\mathcal{G}_{v_1}^1(t)) = \Pr(\mathcal{E}_{v_1}^1(t)) = 1$ for $t \geq 0$.

Induction step: Assume the claim is true for $1 \leq i < n - 1$ and we will prove it is true for $i + 1$. If v^{i+1} is a leaf then we are done since this is an identical case to the base case. Assume v^{i+1} is not a leaf. The case $c = v^{i+1}$ is trivial so consider $c \in C^{i+1}$ that is not v^{i+1} . Then c must reach v^{i+1} via one of its children, let it be v^k where $k < i + 1$. Then by the induction assumption $\Pr(\mathcal{G}_c^k(t')) \geq \Pr(\mathcal{E}_c^k(t'))$ for any t' , and from Lemma 2 for any t the probability that a customer will be served by time t is larger in $\mathcal{N}(G_v, \text{Geom}(p))$ than in $\mathcal{N}(G_v, \text{Exp}(p))$, so we have a faster arrival rate and a faster service rate and the claim follows. ■

C. Proof of Lemma 2

Lemma 2 (restated): *Let X be a geometric random variable with parameter p and supported on the set $\{0, 1, 2, \dots\}$, i.e., for $k \in \mathbb{Z}^+$: $\Pr(X = k) = (1 - p)^k p$, and let Y be an exponential random variable with parameter p . Then, for all $x \in \mathbb{R}^+$:*

$$\Pr(X \leq x) \geq \Pr(Y \leq x) = 1 - e^{-px}, \quad (19)$$

i.e., a random variable $Y \sim \text{Exp}(p)$ stochastically dominates the random variable $X \sim \text{Geom}(p)$.

Proof: For a geometric random variable X with a success probability p and supported on the set $\{0, 1, 2, 3, \dots\}$:

$$\Pr(X > x) = (1 - p)^{x+1}, \text{ for } x \in \mathbb{Z}^+$$

and

$$\Pr(X > x) = (1 - p)^{\lfloor x \rfloor + 1}, \text{ for } x \in \mathbb{R}^+.$$

So, for $x \in \mathbb{R}^+$,

$$\Pr(X \leq x) = 1 - (1 - p)^{\lfloor x \rfloor + 1} \geq 1 - (1 - p)^x = 1 - e^{\ln(1-p)x}$$

and since $\ln(1 - p) \leq -p$ we have:

$$\Pr(X \leq x) \geq 1 - e^{-px}.$$

Hence, if $Y \sim \text{Exp}(p)$ we obtain:

$$\Pr(X \leq x) \geq 1 - e^{-px} = \Pr(Y \leq x),$$

i.e., random variable $Y \sim \text{Exp}(p)$ stochastically dominates the random variable $X \sim \text{Geom}(p)$. ■

D. Proof of Lemma 4

Lemma 4 (restated): *Let Y be the sum of n independent and identically distributed exponential random variables (each one with parameter $\mu > 0$), and $E[Y] = \frac{n}{\mu}$. Then, for $\alpha > 1$:*

$$\Pr(Y < \alpha E[Y]) > 1 - (2e^{-\alpha/2})^n. \quad (20)$$

Proof: The generating function of X is given by:

$$G_X(s) = E[e^{sX}] = \int_0^\infty e^{sx} f_X(x) dx.$$

For any $s < \delta$: $G_X(s) = \frac{\delta}{\delta - s}$. Thus, the generating function of Y (sum of independent X_i 's) for $s < \delta$: $G_Y(s) = (G_X(s))^n = \left(\frac{\delta}{\delta - s}\right)^n$. Now, we will apply a Chernoff bound on Y . For $\delta > s \geq 0$:

$$\begin{aligned} \Pr(Y \geq \alpha E[Y]) &= \Pr\left(Y \geq \alpha \frac{n}{\delta}\right) \\ &= \Pr\left(e^{sY} \geq e^{s \cdot \alpha \frac{n}{\delta}}\right) \leq \frac{E[e^{sY}]}{e^{s \cdot \alpha \frac{n}{\delta}}} = \frac{G_Y(s)}{e^{s \cdot \alpha \frac{n}{\delta}}}. \end{aligned}$$

By letting $s = \delta/2$ we get:

$$\Pr(Y \geq \alpha E[Y]) \leq \left(\frac{\delta}{(\delta - \frac{\delta}{2})e^{\alpha \frac{\delta}{2\delta}}}\right)^n = (2e^{-\alpha/2})^n$$

and thus:

$$\Pr(Y < \alpha E[Y]) > 1 - (2e^{-\alpha/2})^n. \quad \blacksquare$$

E. Proof of the expectation result in Theorem 1

Theorem 1 states that the expected stopping time of algebraic gossip on any graph is: $E[T] = O(\Delta n^2)$ timeslots.

Proof: First, we rewrite the high probability result of Eq. 11 with $\alpha > 1$:

$$\Pr(T \geq 4\alpha n^2 \Delta) \leq 2n(2e^{-\alpha/2})^n.$$

For a positive integer random variable T holds: $E[T] = \sum_{i=1}^{\infty} \Pr(T \geq i)$. So, we have:

$$E[T] = \sum_{i=1}^{\infty} \Pr(T \geq i) \quad (21)$$

$$= \sum_{i=1}^{8n^2\Delta-1} \Pr(T \geq i) + \sum_{i=8n^2\Delta}^{\infty} \Pr(T \geq i) \quad (22)$$

$$\leq 8n^2\Delta + \sum_{i=8n^2\Delta}^{\infty} \Pr(T \geq i) \quad (23)$$

$$\leq 8n^2\Delta + 4n^2\Delta \sum_{\alpha=2}^{\infty} \Pr(T \geq 4\alpha n^2 \Delta). \quad (24)$$

The last inequality is true since $\forall i \leq j, \Pr(T \geq i) \geq \Pr(T \geq j)$ and thus we can replace all $\Pr(T \geq i)$ for $i \in [4\alpha n^2 \Delta, \dots, 4(\alpha+1)n^2 \Delta - 1]$ with $4n^2 \Delta \times \Pr(T \geq 4(\alpha+1)n^2 \Delta)$. Hence,

$$E[T] \leq 8n^2\Delta + 4n^2\Delta \sum_{\alpha=2}^{\infty} \Pr(T \geq 4\alpha n^2 \Delta) \quad (25)$$

$$\leq 8n^2\Delta + 4n^2\Delta \sum_{\alpha=2}^{\infty} 2n(2e^{-\frac{\alpha}{2}})^n \quad (26)$$

$$= 8n^2\Delta + 8n^3\Delta 2^n \sum_{\alpha=2}^{\infty} (e^{-n/2})^\alpha \quad (27)$$

$$= 8n^2\Delta + 8n^3\Delta 2^n \frac{e^{-n}}{1 - e^{-n/2}} \quad (28)$$

$$= 8n^2\Delta + \frac{8n^3\Delta}{1 - e^{-n/2}} \left(\frac{2}{e}\right)^n, \quad (29)$$

$$\text{for } n > 6 : \quad (30)$$

$$\leq 8n^2\Delta + 8n^2\Delta. \quad (31)$$

Thus: $E[T] = O(\Delta n^2)$, and $E[R] = O(\Delta n)$. ■

F. Proof of Lemma 6

Lemma 6 (restated): *Let X be a sum of m independent and identically distributed geometric random variables with parameter p , i.e., $X = \sum_{i=1}^m X_i$. Then, for any positive integer $k < m/p$*

$$\Pr(X > k) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}} kp}\right)^{-m}. \quad (32)$$

Proof: First, we will define Y as the sum of k independent Bernoulli random variables, i.e., $Y = \sum_{i=1}^k Y_i$, where $Y_i \sim \text{Bernoulli}(p)$. Let us notice that:

$$\Pr(X \leq k) = \Pr(Y \geq m)$$

The last is true since the event of observing at least m successes in a sequence of k Bernoulli trials implies that the sum of m independent geometric random variables is no more than k . From the other side, if the sum of m independent geometric random variables is no more than k it implies that m successes occurred not later than the k -th trial and thus $Y \geq m$.

Now we will use a Chernoff bound for the sum of independent Bernoulli random variables presented in [17]: For any $\delta > 0$ and $\mu = E[Y]$:

$$\Pr(Y \geq (1 + \delta)\mu) < \left(\frac{e^\delta}{(1 + \delta)^{1+\delta}}\right)^\mu.$$

Since $\mu = E[Y] = kp$, and by letting $\delta = \frac{m-kp}{kp}$ we obtain:

$$\Pr(Y \geq (1 + \delta)\mu) = \Pr(Y \geq m) < \left(\frac{m}{e^{\frac{m-kp}{m}} kp}\right)^{-m}.$$

So:

$$\Pr(X \leq k) < \left(\frac{m}{e^{\frac{m-kp}{m}} kp}\right)^{-m},$$

and thus the result follows. ■

G. Proof of Theorem 3

For the proof of this theorem we need the following definitions, claims, and lemmas.

1) Stochastic Dominance:

Definition 3 (Stochastic dominance, stochastic ordering [11], [8]). *We say that a random variable X is stochastically less than or equal to a random variable Y if and only if $\Pr(X \leq t) \geq \Pr(Y \leq t)$, and such a relation is denoted as: $X \preceq Y$.*

The proof of the following two claims is omitted.

Claim 1. *If for $i \in \{1, 2\}$, $X_i \preceq Y_i$, X_i are independent, and Y_i are independent, then: $\max_i X_i \preceq \max_i Y_i$.*

Claim 2. *If for $i \in \{1, 2\}$, $X_i \preceq Y_i$, X_i are independent, and Y_i are independent, then: $\sum_i X_i \preceq \sum_i Y_i$.*

2) *Later arrivals yield later departures:* Consider an infinite FCFS queue with a single exponential server. We define a_i as the time of arrival number i to the queue, and d_i as time of the departure number i from the queue. Let X_i be the exponential random variable representing the service time of the arrival i . For all i , X_i 's are *i.i.d.*

Let a_i be a sequence of m arrival times to the queue, and d_i be a sequence of m departure times from the queue.

Lemma 10. *If the sequence a_i is replaced with another sequence of m arrivals \hat{a}_i , such that: $\hat{a}_i \geq a_i \forall i \in [1, \dots, m]$, then, the resulting sequence of m departures will be such that: $\hat{d}_i \geq d_i \forall i \in [1, \dots, m]$. I.e., if every new arrival occurred, stochastically, at the same time or later than the old arrival, then, every new departure from the queue will occur, stochastically, at the same time or later than the old departure.*

Proof: The proof is by induction on the arrival index j , $j \in [1, \dots, m]$.

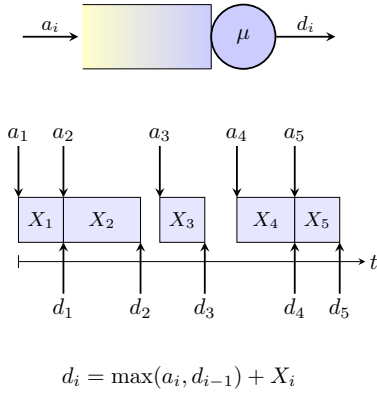


Fig. 4. Arrival and departure times.

- Induction basis: $\hat{d}_1 \succeq d_1$. Follows since $d_1 = a_1 + X_1$, $\hat{d}_1 = \hat{a}_1 + X_1$, and $\hat{a}_1 \succeq a_1$.
- Induction assumption: $\forall i < j : \hat{d}_i \succeq d_i$.
- Induction step: we need to show that $\hat{d}_j \succeq d_j$.

If the j 's arrival occurred when the server was busy, then $d_j = d_{j-1} + X_j$. If the server was idle when the j 's arrival occurred, then $d_j = a_j + X_j$. Thus, we can write:

$$d_j = \max(d_{j-1}, a_j) + X_j, \quad (33)$$

$$\text{and } \hat{d}_j = \max(\hat{d}_{j-1}, \hat{a}_j) + X_j. \quad (34)$$

Since from induction assumption: $\hat{d}_{j-1} \succeq d_{j-1}$, and $\hat{a}_j \succeq a_j$, using Claims 1 and 2, we obtain $\hat{d}_j \succeq d_j$. ■

Proof of Theorem 3:

We denote the nodes of the queuing system Q_n^{tree} as Z_j^l , where l ($l \in [1, \dots, l_{\max}]$) is the level of the node in the tree, and j is the node's index in the level l . The root of the Q_n^{tree} tree is the node Z_1^1 . All servers in the Q_n^{tree} network are ON all the time (work-conserving scheduling), i.e., servers work whenever they have customers to serve. There are no external arrivals to the system. Once a customer is serviced on the level l , it enters the appropriate queue at the level $l-1$. When a customer is serviced by the root Z_1^1 , it leaves the network.

Now, let us define the auxiliary queuing systems: \hat{Q}_n^{tree} and $Q_{l_{\max}}^{line}$.

Definition 4 (Network \hat{Q}_n^{tree}). \hat{Q}_n^{tree} is the same network as Q_n^{tree} with the following change in the servers' scheduling:

At any given moment, only one server at every level l ($l \in [1, \dots, l_{\max}]$) is ON. Once a customer leaves level l , a server that will be scheduled (turned ON) at the level l , is the server which has in its queue a customer that has earliest arrival time to a queue at the level l among all the current customers at the level l . If there are customers that initially reside at the level l , they will be serviced by the order of their IDs (we assume for analysis that every customer has a unique identification number).

Definition 5 (Network of queues $Q_{l_{\max}}^{line}$). $Q_{l_{\max}}^{line}$ is the the following modification of the network \hat{Q}_n^{tree} , that results in a network of l_{\max} queues arranged in a line topology.

For all $l \in [1, \dots, l_{\max}]$, we merge all the nodes at the level l to a single node (a single queue with a single server). We

name this single node at the level l as the first node in Q_n^{tree} at the level l , i.e., Z_1^l . The customers that initially reside at level l will be placed in a single queue in the order of their IDs. This modification results in $Q_{l_{\max}}^{line}$ – a network of l_{\max} queues arranged in a line topology: $Z_1^{l_{\max}} \rightarrow Z_1^{l_{\max}-1} \rightarrow \dots \rightarrow Z_1^1$.

Definition 6 (Network of queues $\hat{Q}_{l_{\max}}^{line}$). $\hat{Q}_{l_{\max}}^{line}$ – is the same system as $Q_{l_{\max}}^{line}$ with the following modification. We take the last customer at some node Z_1^m ($m \in [1, \dots, l_{\max}-1]$) and place it at the head of the queue of the node Z_1^{m+1} . I.e., we move one customer, one queue backward in the line of queues.

Definition 7 (Network of queues $\hat{Q}_{l_{\max}}^{line}$). $\hat{Q}_{l_{\max}}^{line}$ – is the same system as $Q_{l_{\max}}^{line}$ with the following modification. We move all the customers to the queue $Z_1^{l_{\max}}$. I.e., all the customers have to traverse now through all the l_{\max} queues in the line.

We summarize the queuing systems defined above in the short Table I.

Q_n^{tree}	Original system of n queues arranged in a tree topology. Fig. 5 (a).
\hat{Q}_n^{tree}	System of n queues arranged in a tree topology. Only one server is active at each level at a given time. Fig. 5 (b).
$Q_{l_{\max}}^{line}$	System of l_{\max} queues arranged in a line topology. Fig. 5 (c).
$\hat{Q}_{l_{\max}}^{line}$	System of l_{\max} queues arranged in a line topology. One customer is moved one queue backward.
$\hat{Q}_{l_{\max}}^{line}$	System of l_{\max} queues arranged in a line topology. All customers are moved backward to the queue $Z_1^{l_{\max}}$.

TABLE I
QUEUING SYSTEMS USED IN THE PROOF.

The proof of Theorem 3 consists of showing the following relations between the stopping times of the queuing systems:

$$\begin{aligned} t(Q_n^{tree}) &\preceq t(\hat{Q}_n^{tree}) \\ &\approx t(Q_{l_{\max}}^{line}) \\ &\preceq t(\hat{Q}_{l_{\max}}^{line}) \\ &\preceq t(\hat{Q}_{l_{\max}}^{line}) = O(n/\mu). \end{aligned}$$

Stopping time of a queuing system $t(Q)$, is the time by which the last customer leaves the system (via the node Z_1^1). In order to compare the stopping times of queuing systems, we define the following ordered set (or sequence) of departure time from a server Z in a queuing system Q : $d(Z, Q) = (d_1(Z, Q), d_2(Z, Q), \dots, d_i(Z, Q), \dots)$, where $d_i(Z, Q)$ is the time of the departure number i from the node (server) Z .

First, we want to show that the stopping time of Q_n^{tree} is at most the stopping time of the system \hat{Q}_n^{tree} , i.e., $t(Q_n^{tree}) \preceq t(\hat{Q}_n^{tree})$.

Lemma 11. In \hat{Q}_n^{tree} , every departure from the system (via Z_1^1) will occur, stochastically, at the same time or later than in Q_n^{tree} .

$$d_i(Z_1^1, \hat{Q}_n^{tree}) \succeq d_i(Z_1^1, Q_n^{tree}) \quad \forall i \in [1, \dots, n]. \quad (35)$$

Thus, in \hat{Q}_n^{tree} , the last customer will leave the system, stochastically, at the same time or later than in Q_n^{tree} , or: $t(Q_n^{tree}) \preceq t(\hat{Q}_n^{tree})$.

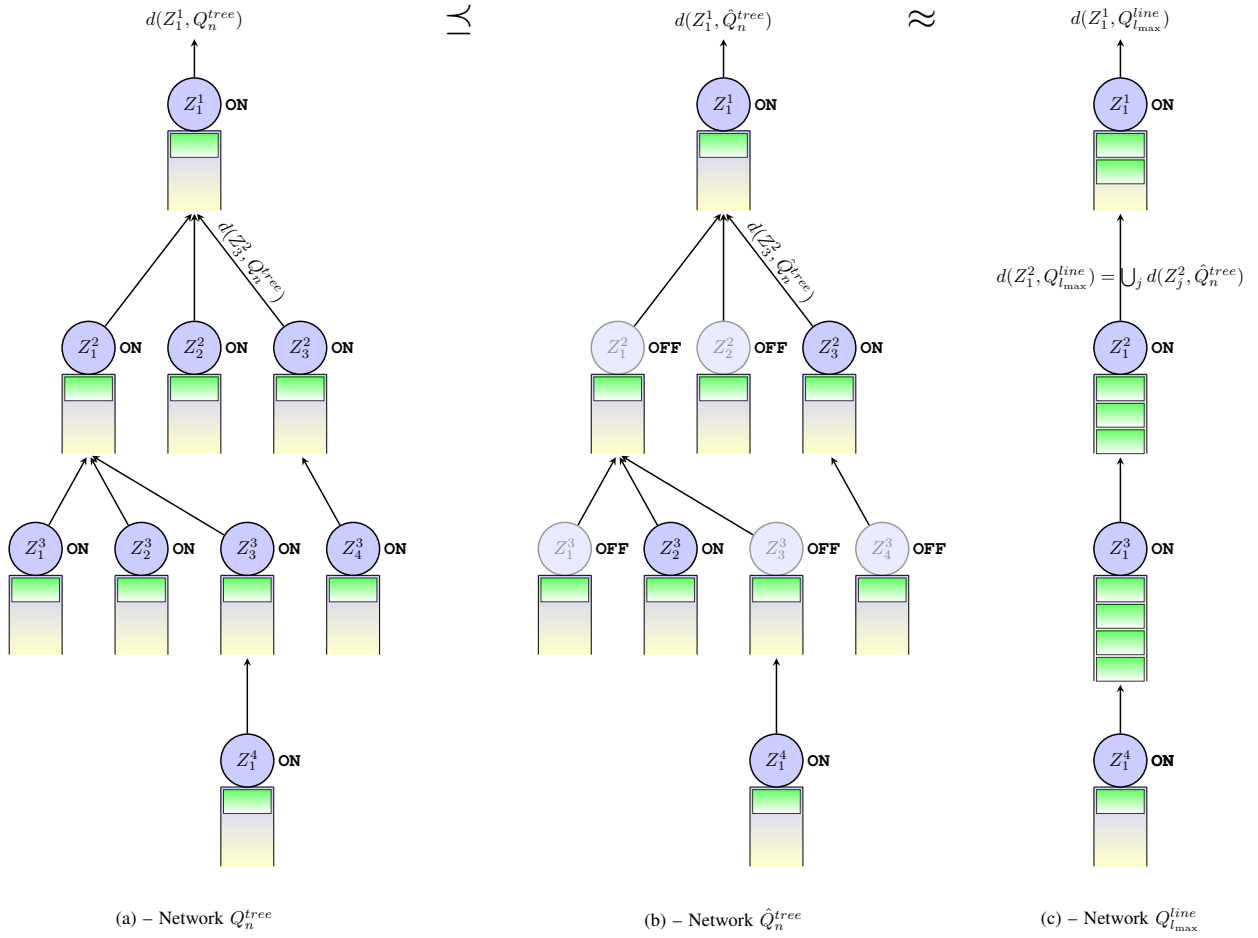


Fig. 5. (a) – Network Q_n^{tree} , where all the servers work all the time. (b) – Network \hat{Q}_n^{tree} , where only one server at each level works at a given time. (c) – Network $Q_{l_{max}}^{line}$.

Proof: The proof is by induction on the tree level l , $l \in [1, \dots, l_{max}]$.

- Induction basis: $\forall i, j : d_i(Z_j^{l_{max}}, \hat{Q}_n^{tree}) \geq d_i(Z_j^{l_{max}}, Q_n^{tree})$. This is true since in \hat{Q}_n^{tree} , the nodes do not work all the time, and thus the departures will occur, stochastically, at the same time or later than in Q_n^{tree} . If there is a single node at the level l_{max} , in \hat{Q}_n^{tree} it will be ON all the time as in Q_n^{tree} , and thus, the departures will occur, stochastically, at the same time in both systems.
- Induction assumption: for all $l > m$ ($m \geq 1$), $\forall i, j : d_i(Z_j^l, \hat{Q}_n^{tree}) \geq d_i(Z_j^l, Q_n^{tree})$.
- Induction step: we need to show that: $\forall i, j : d_i(Z_j^m, \hat{Q}_n^{tree}) \geq d_i(Z_j^m, Q_n^{tree})$.

By induction assumption, for $l = m + 1$: $\forall i, j : d_i(Z_j^{m+1}, \hat{Q}_n^{tree}) \geq d_i(Z_j^{m+1}, Q_n^{tree})$. Now let us take a look at the departures from a node Z_j^m . There are two cases: Z_j^m is a leaf, and Z_j^m is not a leaf. If Z_j^m is a leaf, we can use the same argument as in the induction basis: in \hat{Q}_n^{tree} , the node Z_j^m does not work all the time, and thus the departures from it in \hat{Q}_n^{tree} cannot occur earlier than in Q_n^{tree} . If Z_j^m is not a leaf, it has input/inputs of arrivals from the level $m + 1$. Since the arrivals from the level $m + 1$ in \hat{Q}_n^{tree} occur, stochastically, at the same time or later than in Q_n^{tree}

(by induction assumption), even if the node Z_j^m would work all the time (as in Q_n^{tree}), we would obtain from Lemma 10: $\forall i, j : d_i(Z_j^m, \hat{Q}_n^{tree}) \geq d_i(Z_j^m, Q_n^{tree})$. Moreover, in \hat{Q}_n^{tree} , the node Z_j^m does not work all the time (unless it is the only node at the level m), thus the departure times in \hat{Q}_n^{tree} can be even larger. ■

Lemma 12. In $Q_{l_{max}}^{line}$, every departure from the system (via Z_1^1) will occur, stochastically, at the same time as in \hat{Q}_n^{tree} . Thus, in $Q_{l_{max}}^{line}$, the last customer will leave the system, stochastically, at the same time as in \hat{Q}_n^{tree} .

Proof: Consider the two following facts regarding the network \hat{Q}_n^{tree} . First, a customer entering the level l will be serviced after all the customers that arrived to the level l before it, are serviced. Second, at any given moment, only one customer is being serviced at the level l (if there is at least one customer at the nodes Z_j^l). These facts are true due to the scheduling of the servers in \hat{Q}_n^{tree} (Definition 4).

Clearly, the same facts are true for the network $Q_{l_{max}}^{line}$. First, any customer entering to the level l will be serviced after all the customers that arrived to the level l before it, are serviced. Second, at any given moment, only one customer is being serviced at the level l (if there is at least one customer in the node Z_1^l). These facts are true since in $Q_{l_{max}}^{line}$, at every level,

there is a single queue with a single server (Definition 5).

So, the departure times of every customer from every level l ($l \in [1, \dots, l_{\max}]$) are, stochastically, the same in both systems. The departures from level $l = 1$ are the departures from the node Z_1^1 , and thus the lemma holds. ■

Now we are going to move one customer, one queue backward and will show that the resulting system will have stochastically larger (or the same) stopping time.

Lemma 13. *Consider a network $Q_{l_{\max}}^{line}$. Let m be a level index: $m \in [1, \dots, l_{\max} - 1]$. We take the last customer at the node Z_1^m and place it at the head of the queue of the node Z_1^{m+1} , and call the resulting network – $\hat{Q}_{l_{\max}}^{line}$ (Fig. 6 (b)). Then:*

$$d_i(Z_1^1, Q_{l_{\max}}^{line}) \preceq d_i(Z_1^1, \hat{Q}_{l_{\max}}^{line}) \quad \forall i \in [1, \dots, n]. \quad (36)$$

Thus, in $\hat{Q}_{l_{\max}}^{line}$, the last customer will leave the system, stochastically, at the same time or later than in $Q_{l_{\max}}^{line}$, or: $t(Q_{l_{\max}}^{line}) \preceq t(\hat{Q}_{l_{\max}}^{line})$.

Proof: We call the customer that was moved – customer c . Let us take a look at the times of arrivals to the node Z_1^m in $Q_{l_{\max}}^{line}$ and in $\hat{Q}_{l_{\max}}^{line}$. Since the customer c is already located in the queue of Z_1^m in $Q_{l_{\max}}^{line}$, its arrival time can be considered as 0. In $\hat{Q}_{l_{\max}}^{line}$, the arrival time of c is at least 0 (it should be serviced at Z_1^{m+1} before arriving at Z_1^m). Each one of the rest customers that should arrive at Z_1^m will arrive in $\hat{Q}_{l_{\max}}^{line}$, stochastically, at the same time or later than in $Q_{l_{\max}}^{line}$, since in $\hat{Q}_{l_{\max}}^{line}$ the server Z_1^{m+1} should first service the customer c , and only then will start servicing the rest customers. Thus, $d_i(Z_1^{m+1}, \hat{Q}_{l_{\max}}^{line}) \succeq d_i(Z_1^{m+1}, Q_{l_{\max}}^{line})$. Using Lemma 10 we obtain that: $d_i(Z_1^m, \hat{Q}_{l_{\max}}^{line}) \succeq d_i(Z_1^m, Q_{l_{\max}}^{line})$. Iteratively applying Lemma 10 to the nodes Z_1^l , $l \in [m - 1, \dots, 1]$, we obtain the result: $d_i(Z_1^1, \hat{Q}_{l_{\max}}^{line}) \succeq d_i(Z_1^1, Q_{l_{\max}}^{line})$. ■

Corollary 4. *Consider a network $\hat{Q}_{l_{\max}}^{line}$ (Definition 7) which is identical to the network $Q_{l_{\max}}^{line}$ with the following change. In $\hat{Q}_{l_{\max}}^{line}$, all the n customers are located at the node $Z_1^{l_{\max}}$ (Fig. 6 (c)). Then:*

$$d_i(Z_1^1, Q_{l_{\max}}^{line}) \preceq d_i(Z_1^1, \hat{Q}_{l_{\max}}^{line}) \quad \forall i \in [1, \dots, n]. \quad (37)$$

Thus, in $\hat{Q}_{l_{\max}}^{line}$, the last customer will leave the system, stochastically, at the same time or later than in $Q_{l_{\max}}^{line}$, or: $t(Q_{l_{\max}}^{line}) \preceq t(\hat{Q}_{l_{\max}}^{line})$.

Proof: Given the network $Q_{l_{\max}}^{line}$ we take one customer from the tail of some queue (except the queue of the node $Z_1^{l_{\max}}$) and place it at the head of the queue of the preceding node in the $Q_{l_{\max}}^{line}$. According to the Lemma 13, we get a network in which every customer leaves via Z_1^1 , stochastically, not earlier than in $Q_{l_{\max}}^{line}$. Iteratively moving customers (one customer and one queue at a time) backwards we get finally the network $\hat{Q}_{l_{\max}}^{line}$ in which all the n customers are located at the node $Z_1^{l_{\max}}$. Since at each step, according to Lemma 13, the departure times from Z_1^1 could only get, stochastically, larger, the lemma holds. ■

Corollary 5. *The time it will take the last customer to leave the network of n queues arranged in a tree topology is,*

stochastically, the same or smaller than in the network of n queues arranged in a line topology where all the n customers are located at the farthest queue, i.e., $t(Q_n^{tree}) \preceq t(\hat{Q}_{l_{\max}}^{line})$.

Proof: This corollary is a direct consequence of the Lemmas 11, 12, and the Corollary 4. ■

Now we are ready for the last step of the proof. We will find the stopping time of a system of queues arranged in a line topology and with all the customers located at the last queue.

Lemma 14. *The time it will take to the last customer to leave the system $\hat{Q}_{l_{\max}}^{line}$ (l_{\max} queues arranged in a line topology) is $O(n/\mu)$ with high probability. Formally, for any $\alpha > 1$:*

$$\Pr(t(\hat{Q}_{l_{\max}}^{line}) < \alpha 4n/\mu) > 1 - 2(2e^{-\alpha/2})^n. \quad (38)$$

Proof: Initially, all the customers (from now we will call them *real* customers) are located in the last ($Z_1^{l_{\max}}$) queue. We now take all the *real* customers out of this queue and will make them enter the system (via the $Z_1^{l_{\max}}$) from outside. We define the *real* customers' arrivals as a Poisson process with rate $\lambda = \frac{\mu}{2}$. So, $\rho = \frac{\lambda}{\mu} = \frac{1}{2} < 1$ for all the queues in the system. Clearly, such an assumption only increases the stopping time of the system (stopping time is the time until the last customer leaves the system). According to Jackson's theorem, which proof can be found in [5], there exists an equilibrium state. So, we need to ensure that the lengths of all queues at time $t = 0$ are according to the equilibrium state probability distribution. We add *dummy* customers to all the queues according to the stationary distribution. By adding additional *dummy* customers to the system, we make the *real* customers wait longer in the queues, thus increasing the stopping time.

We will compute the stopping time $t(\hat{Q}_{l_{\max}}^{line})$ in two phases: Let us denote this time as $t_1 + t_2$, where t_1 is the time needed for the n 'th customer to arrive at the first queue, and t_2 is the time needed for the n 'th customer to pass through all the l_{\max} queues in the system.

From Jackson's Theorem, it follows that the number of customers in each queue is independent, which implies that the random variables that represent the waiting times in each queue are independent.

The random variable t_1 is the sum of n independent random variables distributed exponentially with parameter $\mu/2$. From Lemma 3 we obtain that t_2 is the sum of l_{\max} independent random variables distributed exponentially with parameter $\mu - \lambda = \mu/2$ (Lemma 3). Since $l_{\max} \leq n$, we can assume the worst case (for the upper bound of stopping time) $l_{\max} = n$. Thus, we can view t_2 as the sum of n independent random variables distributed exponentially with parameter $\mu/2$. $E[t_1] = \sum_{i=1}^n 2/\mu = 2n/\mu$, so, using Lemma 4:

$$\Pr(t_1 < \alpha E[t_1]) > 1 - (2e^{-\alpha/2})^n, \quad (39)$$

$$\Pr(t_1 < \alpha 2n/\mu) > 1 - (2e^{-\alpha/2})^n. \quad (40)$$

In a similar way we obtain:

$$\Pr(t_2 < \alpha 2n/\mu) > 1 - (2e^{-\alpha/2})^n. \quad (41)$$

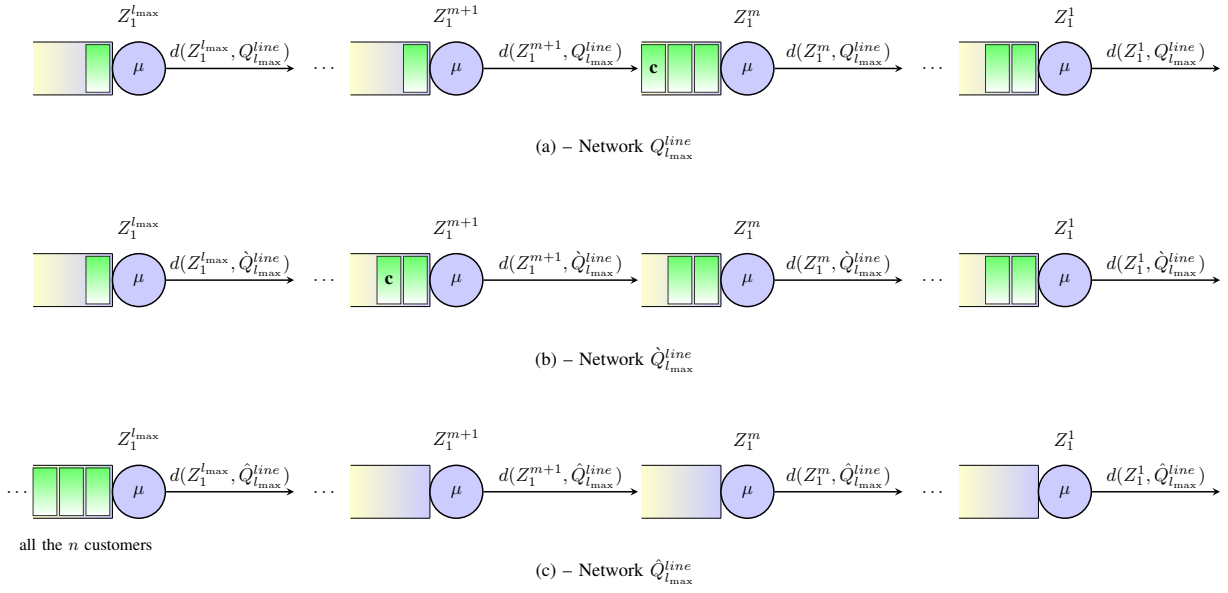


Fig. 6. (a) – Network $Q_{l_{\max}}^{line}$. (b) – Network \hat{Q}_n^{tree} , where one customer is moved one queue backward. (c) – Network $\hat{Q}_{l_{\max}}^{line}$, where all the customers are at the last queue.

$t(\hat{Q}_{l_{\max}}^{line}) = t_1 + t_2$, thus, using union bound:

$$\Pr(t_1 + t_2 < \alpha 4n/\mu) > 1 - 2(2e^{-\alpha/2})^n. \quad (42)$$

and thus:

$$t(\hat{Q}_{l_{\max}}^{line}) = O(n/\mu) \text{ (for a constant } \alpha) \quad (43)$$

w.p. of at least $1 - 2(2e^{-\alpha/2})^n$.

From Claim 5 we obtain that $t(Q_n^{tree}) \leq t(\hat{Q}_{l_{\max}}^{line})$ and thus:
 $t(Q_n^{tree}) < \alpha 4n/\mu$ w.p. of at least $1 - 2(2e^{-\alpha/2})^n$ for any $\alpha > 1$, so the proof of Theorem 3 is completed. ■