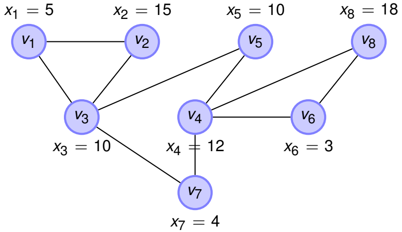
Tight Bounds for Algebraic Gossip on Graphs

Michael Borokhovich   Chen Avin   Zvi Lotker

Department of Communication Systems Engineering
Ben-Gurion University of the Negev, Israel

ISIT 2010

The Task: Disseminate All Values to All Nodes



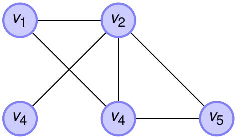
- Every node needs to learn all the values.
- A node knows only its neighbors.
- A node does not know what values other nodes know.
- We are looking for a local and distributed algorithm.

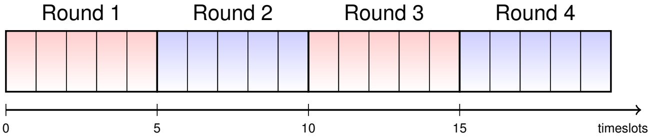
Time Model

- Time is discrete and divided into timeslots.

Time Model

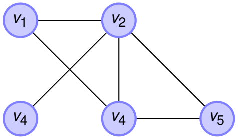
- Time is discrete and divided into timeslots.
- n consecutive timeslots are regarded as a round.

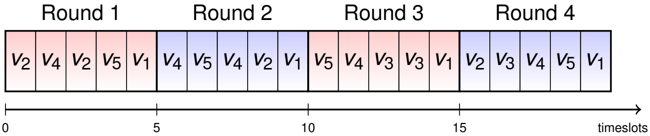




Time Model

- Time is discrete and divided into timeslots.
- n consecutive timeslots are regarded as a round.

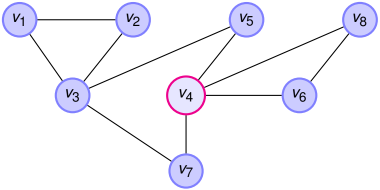




- Every timeslot - **random** node wakes up for a **gossip action**.

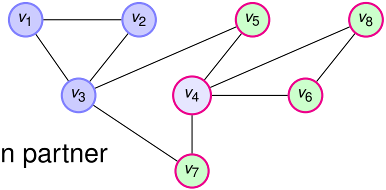
Gossip Algorithm – The Way Information is Spread

At each timeslot a **single random node** takes a **gossip action**



Gossip Algorithm – The Way Information is Spread

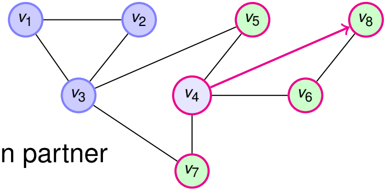
At each timeslot a **single random node** takes a **gossip action**



- Gossip Algorithm:
  1. Determines a communication partner randomly among neighbors.
     - Uniform gossip.
     - Non uniform gossip.

Gossip Algorithm – The Way Information is Spread

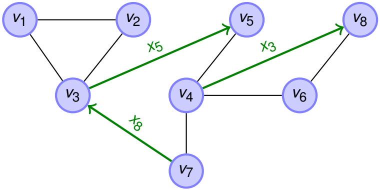
At each timeslot a **single random node** takes a **gossip action**



- Gossip Algorithm:
  1. Determines a communication partner randomly among neighbors.
     - Uniform gossip.
     - Non uniform gossip.
  2. Determines how the message is sent.
     - **PUSH** – a message is sent to the partner.
     - **PULL** – a message is sent from the partner.
     - **EXCHANGE** - PUSH and PULL.

Algebraic Gossip is Based on Random Linear Network Coding

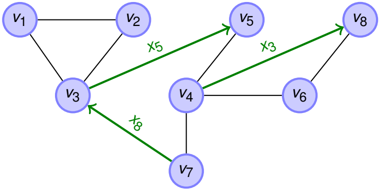
instead of sending randomly chosen values...



every message – a single value

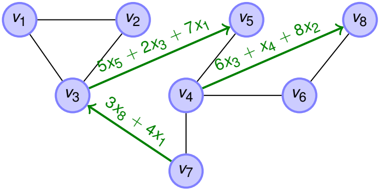
Algebraic Gossip is Based on Random Linear Network Coding

instead of sending randomly chosen values...



every message – a single value

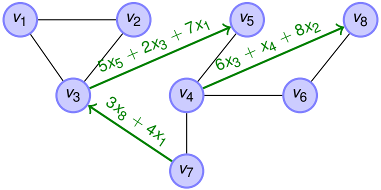
we send random linear combinations



every message – linear equation

Algebraic Gossip is Based on Random Linear Network Coding

we send random linear combinations



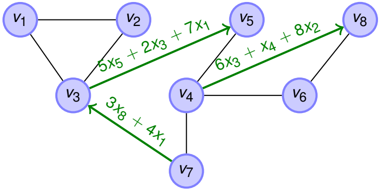
every message – linear equation

linear equations are stored in a matrix form:

$$\begin{bmatrix} 4 & 3 & 7 & 6 & 3 & 0 & 3 & 1 \\ 2 & 0 & 0 & 7 & 4 & 0 & 5 & 0 \\ 1 & 1 & 0 & 0 & 3 & 3 & 0 & 0 \\ 0 & 2 & 1 & 5 & 4 & 0 & 0 & 0 \\ 4 & 0 & 7 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 22 \\ 45 \\ 78 \\ 30 \\ 15 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Algebraic Gossip is Based on Random Linear Network Coding

we send random linear combinations



every message – linear equation

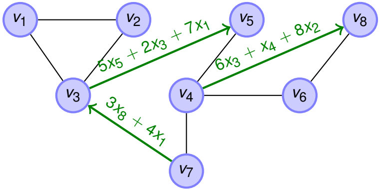
linear equations are stored in a matrix form:

$$\begin{bmatrix} 4 & 3 & 7 & 6 & 3 & 0 & 3 & 1 \\ 2 & 0 & 0 & 7 & 4 & 0 & 5 & 0 \\ 1 & 1 & 0 & 0 & 3 & 3 & 0 & 0 \\ 0 & 2 & 1 & 5 & 4 & 0 & 0 & 0 \\ 4 & 0 & 7 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 22 \\ 45 \\ 78 \\ 30 \\ 15 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

once a node has rank n – it finishes

Algebraic Gossip is Based on Random Linear Network Coding

we send random linear combinations



every message – linear equation

linear equations are stored in a matrix form:

$$\begin{bmatrix} 4 & 3 & 7 & 6 & 3 & 0 & 3 & 1 \\ 2 & 0 & 0 & 7 & 4 & 0 & 5 & 0 \\ 1 & 1 & 0 & 0 & 3 & 3 & 0 & 0 \\ 0 & 2 & 1 & 5 & 4 & 0 & 0 & 0 \\ 4 & 0 & 7 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 22 \\ 45 \\ 78 \\ 30 \\ 15 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

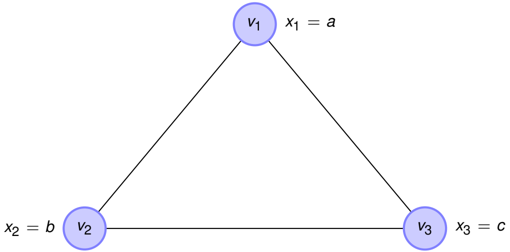
once a node has rank n – it finishes

only **helpful** messages are stored

messages that increase the rank of the matrix

Algebraic Gossip – Simple Example

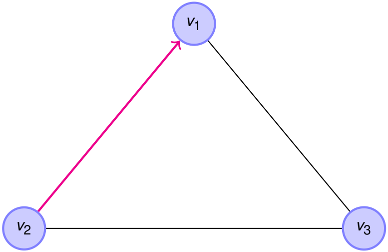
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$

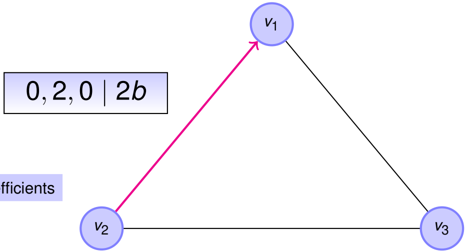


random coefficients

$$\begin{matrix} 3 \\ 2 \\ 5 \end{matrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$

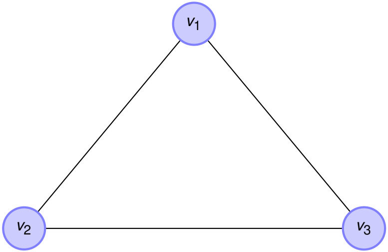


random coefficients

$$\begin{matrix} 3 \\ 2 \\ 5 \end{matrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 0 \end{bmatrix}$$

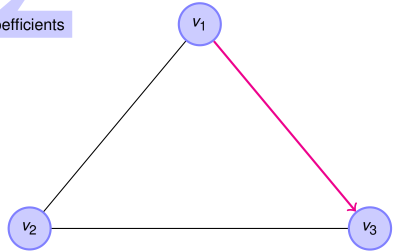


$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{matrix} 5 \\ 6 \\ 2 \end{matrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 0 \end{bmatrix}$$

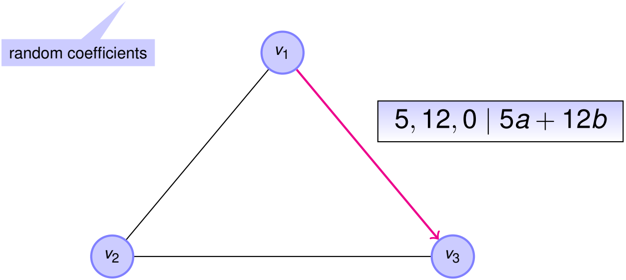
random coefficients



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

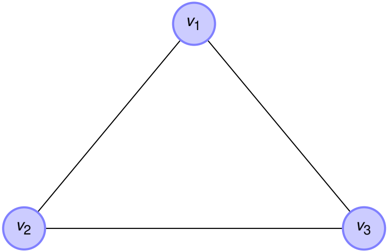
$$\begin{matrix}5\\6\\2\end{matrix}\begin{bmatrix}1 & 0 & 0\\0 & 2 & 0\\0 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}a\\2b\\0\end{bmatrix}$$



$$\begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}0\\b\\0\end{bmatrix} \qquad \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}0\\0\\c\end{bmatrix}$$

Algebraic Gossip – Simple Example

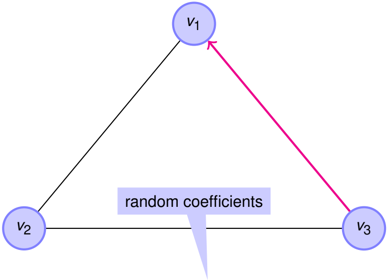
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 5 & 12 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5a + 12b \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 0 \end{bmatrix}$$

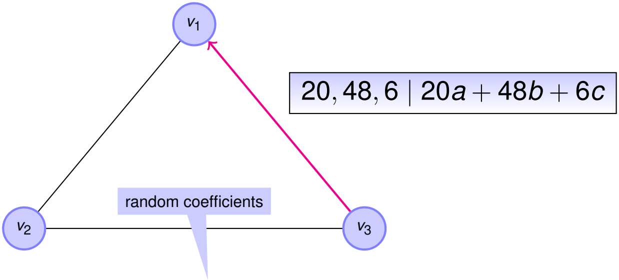


$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}$$

$$\begin{matrix} 4 \\ 3 \\ 6 \end{matrix} \begin{bmatrix} 5 & 12 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5a + 12b \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

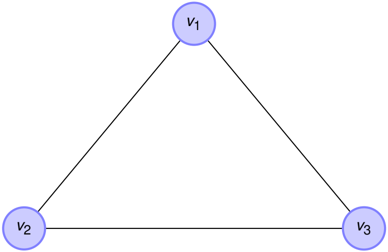
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{matrix} 4 \\ 3 \\ 6 \end{matrix} \begin{bmatrix} 5 & 12 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5a + 12b \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 20 & 48 & 6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 20a + 48b + 6c \end{bmatrix}$$

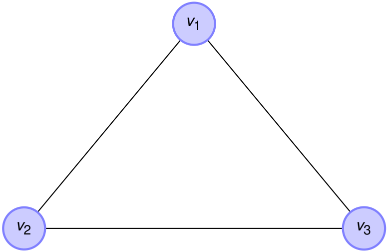


$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 5 & 12 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5a + 12b \\ 0 \\ c \end{bmatrix}$$

Algebraic Gossip – Simple Example

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 20 & 48 & 6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a \\ 2b \\ 20a + 48b + 6c \end{bmatrix}$$

$$\Rightarrow \begin{aligned} x_1 &= a \\ x_2 &= b \\ x_3 &= c \end{aligned}$$

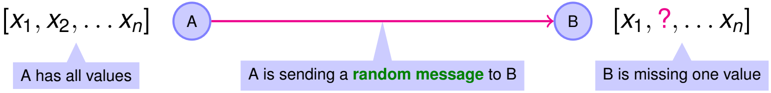


$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 12 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5a + 12b \\ 0 \\ c \end{bmatrix}$$

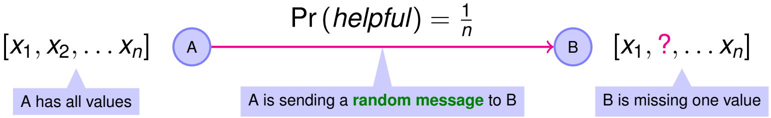
So, Why Algebraic Gossip is Faster?

Without Algebraic Gossip (Random Message Selection)



So, Why Algebraic Gossip is Faster?

Without Algebraic Gossip (Random Message Selection)



So, Why Algebraic Gossip is Faster?

Without Algebraic Gossip (Random Message Selection)

$[x_1, x_2, \ldots x_n]$ A → B $[x_1, ?, \ldots x_n]$

$\Pr(helpful) = \frac{1}{n}$

A has all values

A is sending a **random message** to B

B is missing one value

With Algebraic Gossip (Random Linear Equations)

[rank n] A → B [rank $n-1$]

A has n independent equations

A is sending a **random linear equation** to B

B has $n-1$ independent equations

So, Why Algebraic Gossip is Faster?

Without Algebraic Gossip (Random Message Selection)

$[x_1, x_2, \ldots x_n]$ A → B $[x_1, ?, \ldots x_n]$

$\Pr(helpful) = \frac{1}{n}$

A has all values

A is sending a **random message** to B

B is missing one value

With Algebraic Gossip (Random Linear Equations)

[rank n] A → B [rank $n - 1$]

$\Pr(helpful) = 1 - \frac{1}{q}$

A has n independent equations

A is sending a **random linear equation** to B

B has $n - 1$ independent equations

So, Why Algebraic Gossip is Faster?

Without Algebraic Gossip (Random Message Selection)

$$[x_1, x_2, \ldots x_n] \quad A \xrightarrow{\Pr(helpful) = \frac{1}{n}} B \quad [x_1, ?, \ldots x_n]$$

- A has all values
- A is sending a **random message** to B
- B is missing one value

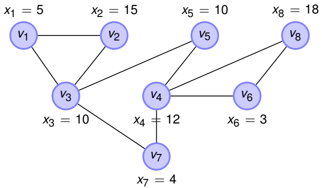
With Algebraic Gossip (Random Linear Equations)

$$[\text{rank } n] \quad A \xrightarrow{\Pr(helpful) = 1 - \frac{1}{q}} B \quad [\text{rank } n-1]$$

- A has n independent equations
- A is sending a **random linear equation** to B
- B has $n-1$ independent equations

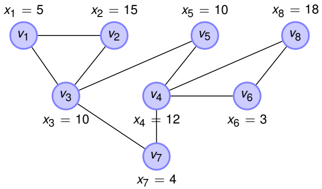
$$\Pr(helpful) = \frac{q^n - q^{n-1}}{q^n} = 1 - \frac{1}{q}$$

Research Goal – Stopping Time of Algebraic Gossip



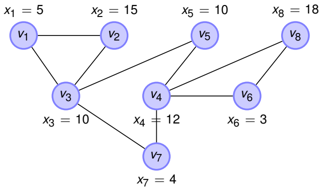
- Given an arbitrary graph with *n* nodes, how many communication rounds are needed to complete the *algebraic gossip* algorithm?

Research Goal – Stopping Time of Algebraic Gossip



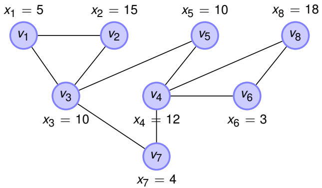
- Given an arbitrary graph with n nodes, how many communication rounds are needed to complete the *algebraic gossip* algorithm?
- Known results:
  - [Deb et al., 2006] – Tight bound for complete graph.

Research Goal – Stopping Time of Algebraic Gossip



- Given an arbitrary graph with n nodes, how many communication rounds are needed to complete the *algebraic gossip* algorithm?
- Known results:
  - [Deb et al., 2006] – Tight bound for complete graph.
  - [Mosk-Aoyama and Shah, 2006] – Upper bound for arbitrary graphs, based on conductance measure. The bound is not tight.

Research Goal – Stopping Time of Algebraic Gossip



- Given an arbitrary graph with n nodes, how many communication rounds are needed to complete the *algebraic gossip* algorithm?
- Known results:
  - [Deb et al., 2006] – Tight bound for complete graph.
  - [Mosk-Aoyama and Shah, 2006] – Upper bound for arbitrary graphs, based on conductance measure. The bound is not tight.
  - Worst case lower bound was not addressed before.

Our Main Result – Tight Bound for Algebraic Gossip

Theorem 1: Upper Bound

For any graph with n nodes and maximum degree Δ, stopping time of algebraic gossip is $O(\Delta n)$ in expectation and with high probability.

Our Main Result – Tight Bound for Algebraic Gossip

Theorem 1: Upper Bound

For any graph with n nodes and maximum degree Δ,
stopping time of algebraic gossip is $O(\Delta n)$
in expectation and with high probability.

Corollary 1

For any graph with n nodes,
stopping time of algebraic gossip is $O(n^2)$.

Our Main Result – Tight Bound for Algebraic Gossip

Theorem 1: Upper Bound

For any graph with n nodes and maximum degree Δ,
stopping time of algebraic gossip is $O(\Delta n)$
in expectation and with high probability.

Corollary 1

For any graph with n nodes,
stopping time of algebraic gossip is $O(n^2)$.

Corollary 2

For any graph with n nodes and
a **constant maximum degree** Δ
stopping time of algebraic gossip is $\Theta(n)$.

Our Main Result – Tight Bound for Algebraic Gossip

Theorem 1: Upper Bound

For any graph with n nodes and maximum degree Δ, stopping time of algebraic gossip is $O(\Delta n)$ in expectation and with high probability.

Theorem 2: Worst Case Lower Bound

For any $2 \leq \Delta \leq \frac{n}{2} + 1$, there exists a graph with maximum degree Δ, for which algebraic gossip takes $\Omega(\Delta n)$ rounds.

Our Main Result – Tight Bound for Algebraic Gossip

Theorem 1: Upper Bound

For any graph with n nodes and maximum degree Δ, stopping time of algebraic gossip is $O(\Delta n)$ in expectation and with high probability.

Theorem 2: Worst Case Lower Bound

For any $2 \leq \Delta \leq \frac{n}{2} + 1$, there exists a graph with maximum degree Δ, for which algebraic gossip takes $\Omega(\Delta n)$ rounds.

Corollary 3

There exists a graph with n nodes, for which stopping time of algebraic gossip is $\Omega(n^2)$.

Proof Overview – Very Important Lemma

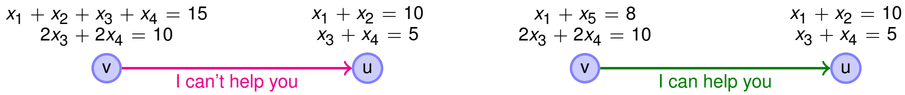
$$x_1 + x_2 + x_3 + x_4 = 15 \qquad x_1 + x_2 = 10$$
$$2x_3 + 2x_4 = 10 \qquad x_3 + x_4 = 5$$

$$x_1 + x_5 = 8 \qquad x_1 + x_2 = 10$$
$$2x_3 + 2x_4 = 10 \qquad x_3 + x_4 = 5$$

Definition

A node v is called **helpful** to a node u,
if it can build a linear equation that is
linearly independent of all equations that u has.

Proof Overview – Very Important Lemma



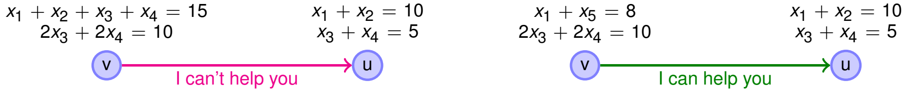
Definition

A node v is called **helpful** to a node u,
if it can build a linear equation that is
linearly independent of all equations that u has.

Lemma [Deb et al., 2006]

If a node v is **helpful** to a node u,
it will actually help it by sending a message,
with probability of at least $1 - \frac{1}{q}$.

Proof Overview – Very Important Lemma



Definition

A node v is called **helpful** to a node u,
if it can build a linear equation that is
linearly independent of all equations that u has.

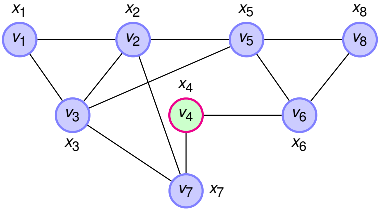
Lemma [Deb et al., 2006]

If a node v is **helpful** to a node u,
it will actually help it by sending a message,
with probability of at least $1 - \frac{1}{q}$.

q is a field size from which random coefficients are drawn

Proof Overview – Converting a Graph to a System of Queues

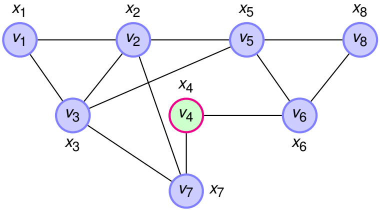
initial graph with an arbitrary node v_4



when v_4 will finish the protocol?

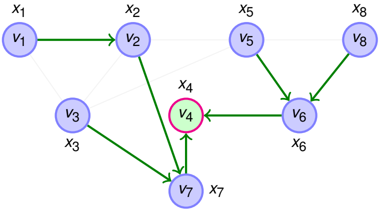
Proof Overview – Converting a Graph to a System of Queues

initial graph with an arbitrary node v_4



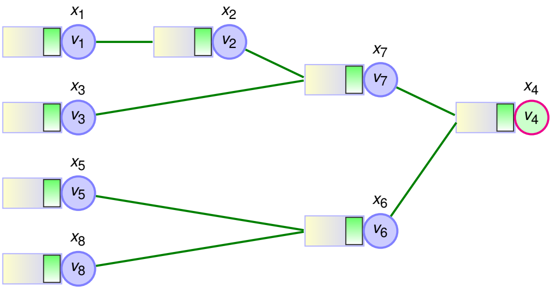
when v_4 will finish the protocol?

BFS spanning tree rooted at v_4



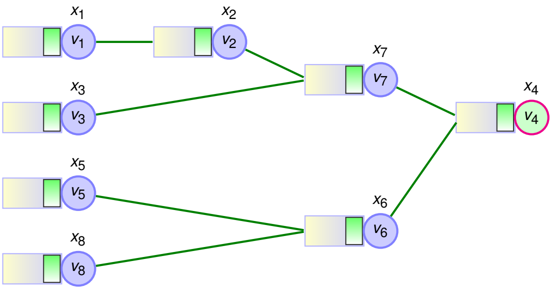
we **ignore** the messages coming from other edges

Proof Overview – Converting a Graph to a System of Queues



customers are **helpful** messages

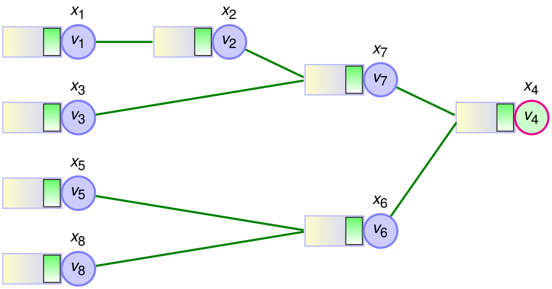
Proof Overview – Converting a Graph to a System of Queues



customers are **helpful** messages

initially, every node has one **helpful** message

Proof Overview – Converting a Graph to a System of Queues

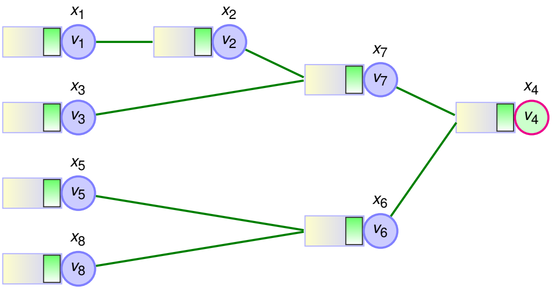


customers are **helpful** messages

initially, every node has one **helpful** message

customer arriving at some node, increases its rank by 1

Proof Overview – Converting a Graph to a System of Queues



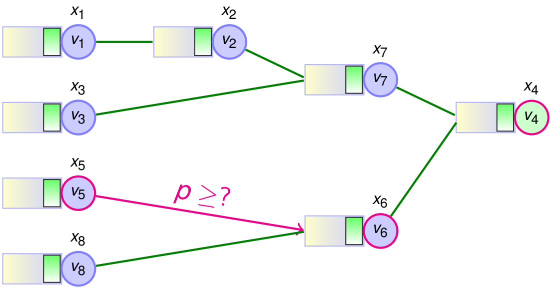
customers are **helpful** messages

initially, every node has one **helpful** message

customer arriving at some node, increases its rank by 1

once v_4 receives n **helpful** messages it finishes

Proof Overview – Converting a Graph to a System of Queues



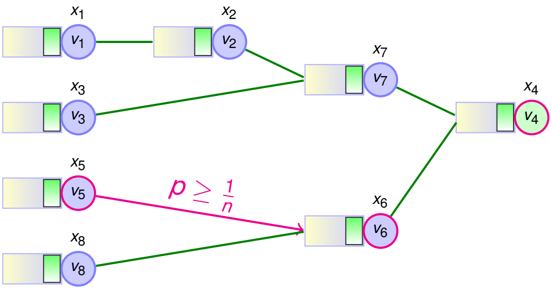
customers are **helpful** messages

initially, every node has one **helpful** message

customer arriving at some node, increases its rank by 1

once v_4 receives n **helpful** messages it finishes

Proof Overview – Converting a Graph to a System of Queues



customers are **helpful** messages

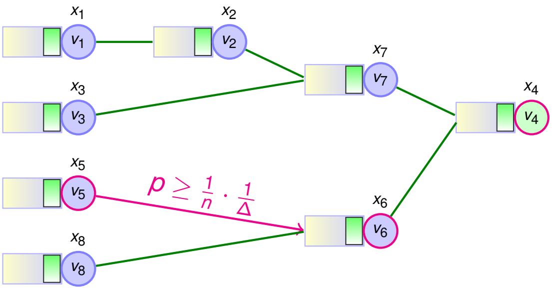
initially, every node has one **helpful** message

customer arriving at some node, increases its rank by 1

once v_4 receives n **helpful** messages it finishes

in a given **timeslot**, v_5 wakes up w.p. $\frac{1}{n}$

Proof Overview – Converting a Graph to a System of Queues



customers are **helpful** messages

initially, every node has one **helpful** message

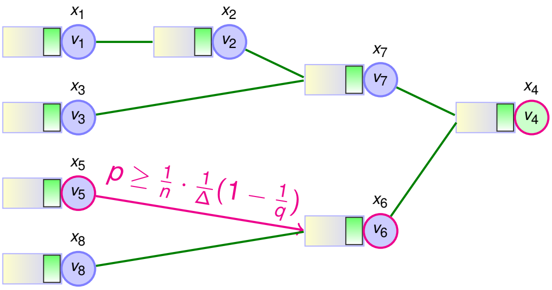
customer arriving at some node, increases its rank by 1

once v_4 receives n **helpful** messages it finishes

in a given **timeslot**, v_5 wakes up w.p. $\frac{1}{n}$

v_5 chooses v_6 as a partner w.p. $\geq \frac{1}{\Delta}$

Proof Overview – Converting a Graph to a System of Queues



customers are **helpful** messages

initially, every node has one **helpful** message

customer arriving at some node, increases its rank by 1

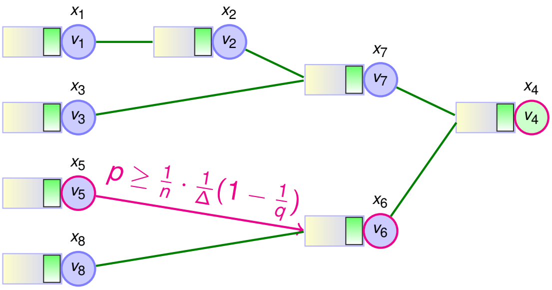
once v_4 receives n **helpful** messages it finishes

in a given **timeslot**, v_5 wakes up w.p. $\frac{1}{n}$

v_5 chooses v_6 as a partner w.p. $\geq \frac{1}{\Delta}$

the message will be **helpful** w.p. $\geq (1 - \frac{1}{q})$

Proof Overview – Converting a Graph to a System of Queues



customers are **helpful** messages

initially, every node has one **helpful** message

customer arriving at some node, increases its rank by 1

once v_4 receives n **helpful** messages it finishes

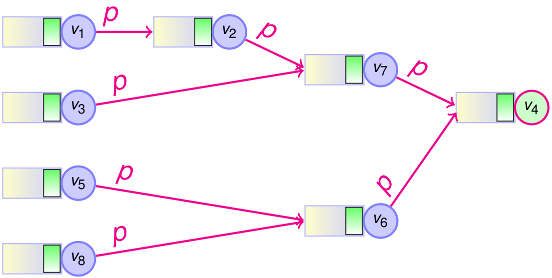
in a given **timeslot**, v_5 wakes up w.p. $\frac{1}{n}$

v_5 chooses v_6 as a partner w.p. $\geq \frac{1}{\Delta}$

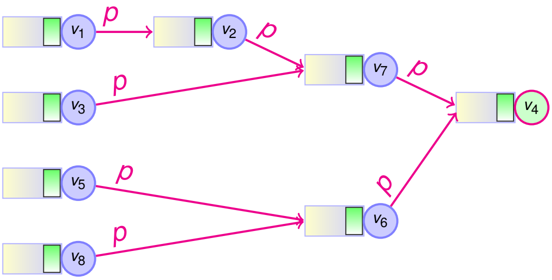
the message will be **helpful** w.p. $\geq (1 - \frac{1}{q})$

service time is geometrically distributed with p

Proof Overview – Exponential Servers Instead of Geometric

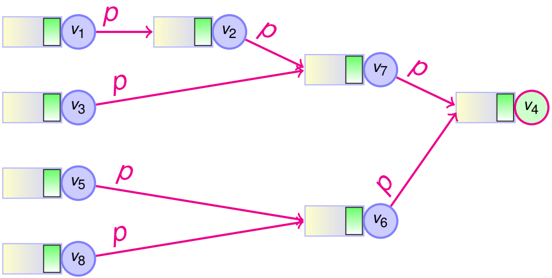


Proof Overview – Exponential Servers Instead of Geometric



If $X \sim \text{Geom}(p)$, and $Y \sim \text{Exp}(p)$, then: $\Pr(Y > t) \geq \Pr(X > t)$

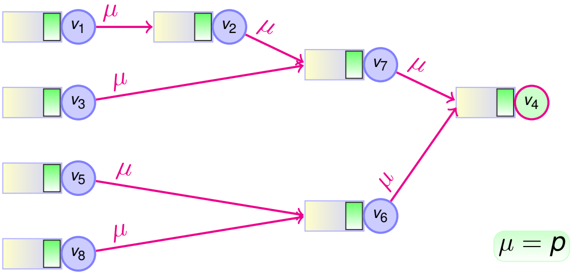
Proof Overview – Exponential Servers Instead of Geometric



If $X \sim \text{Geom}(p)$, and $Y \sim \text{Exp}(p)$, then: $\Pr(Y > t) \geq \Pr(X > t)$

so, exponential server is slower than geometric

Proof Overview – Exponential Servers Instead of Geometric

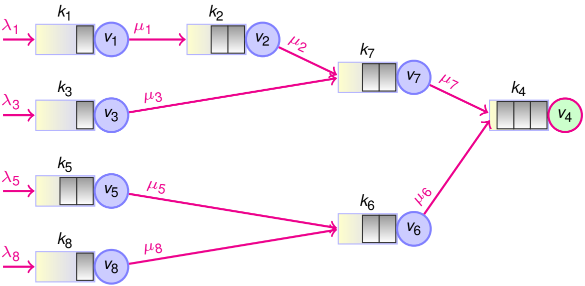


If $X \sim \text{Geom}(p)$, and $Y \sim \text{Exp}(p)$, then: $\Pr(Y > t) \geq \Pr(X > t)$

so, exponential server is slower than geometric

we replace servers, thus increasing the stopping time

Proof Overview – Jackson's Theorem

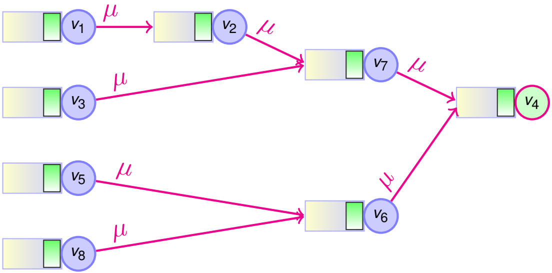


- **Jackson's Theorem:** If utilization at every queue is less than 1, the equilibrium state distribution of number of customers in each queue exists and it is given by:

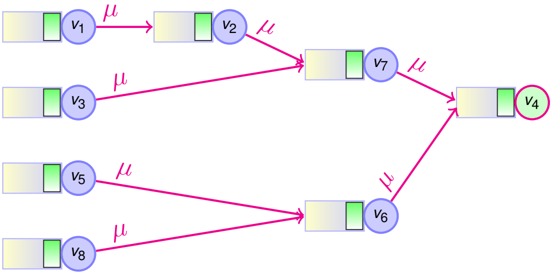
For state $(k_1, k_2, \ldots, k_n)$, and utilization $\rho_i = \frac{\lambda_i}{\mu_i}$:

$$\pi(k_1, k_2, \ldots, k_n) = \prod_{i=1}^{n} \rho_i^{k_i}(1 - \rho_i).$$

Proof Overview – Applying Jackson's Theorem

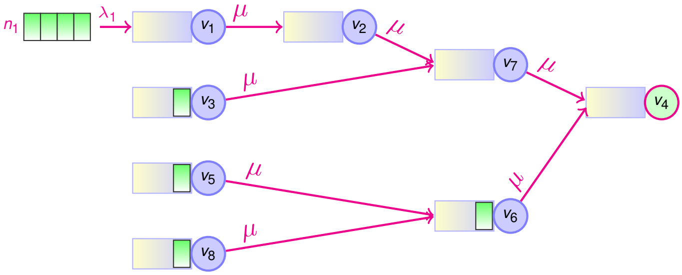


Proof Overview – Applying Jackson's Theorem



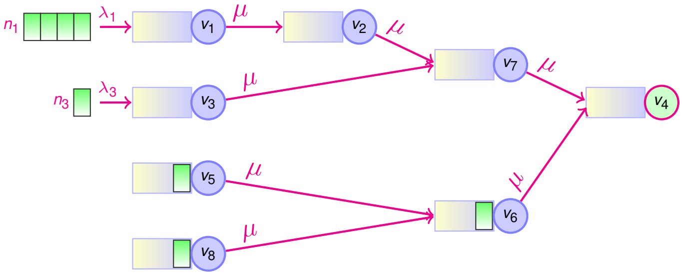
we take all customers out of the system

Proof Overview – Applying Jackson's Theorem



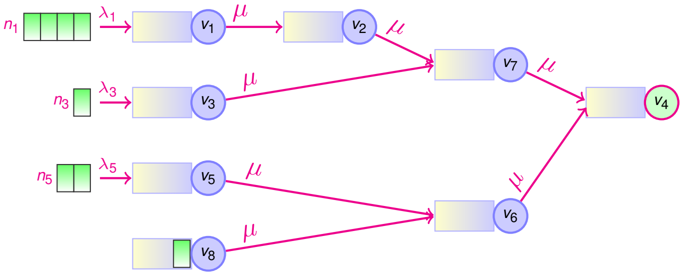
we take all customers out of the system

Proof Overview – Applying Jackson's Theorem



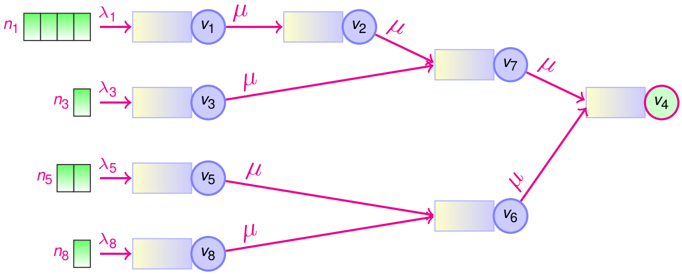
we take all customers out of the system

Proof Overview – Applying Jackson's Theorem



we take all customers out of the system

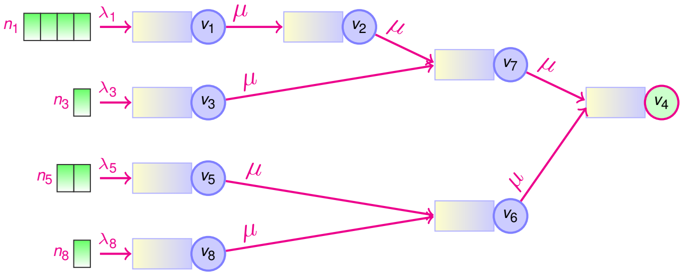
Proof Overview – Applying Jackson's Theorem



we take all customers out of the system

every customer will traverse through additional queues

Proof Overview – Applying Jackson's Theorem

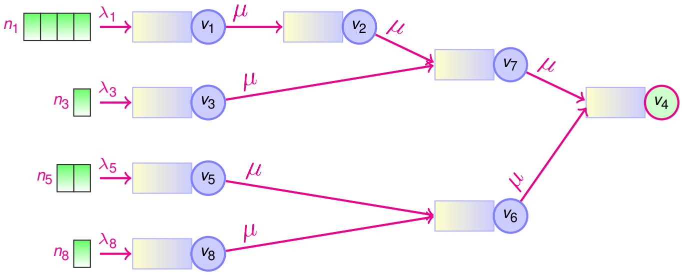


we take all customers out of the system

every customer will traverse through additional queues

by setting: $\lambda_i = \frac{\mu n_i}{2n}$, we obtain: $\rho_i < 1$

Proof Overview – Applying Jackson's Theorem



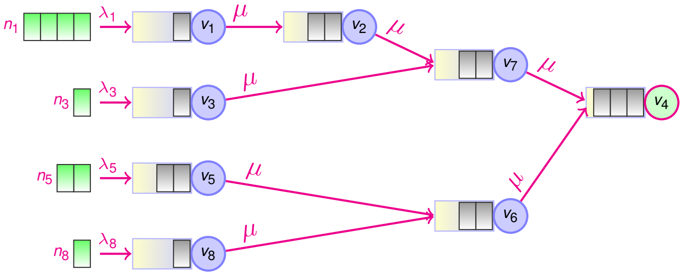
we take all customers out of the system

every customer will traverse through additional queues

by setting: $\lambda_i = \frac{\mu n_i}{2n}$, we obtain: $\rho_i < 1$

according to Jackson, stationary distribution exists

Proof Overview – Applying Jackson's Theorem



we take all customers out of the system

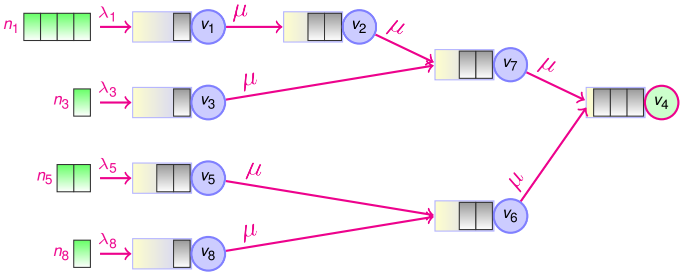
every customer will traverse through additional queues

by setting: $\lambda_i = \frac{\mu n_i}{2n}$, we obtain: $\rho_i < 1$

according to Jackson, stationary distribution exists

we add **dummy customers**

Proof Overview – Applying Jackson's Theorem



we take all customers out of the system

every customer will traverse through additional queues

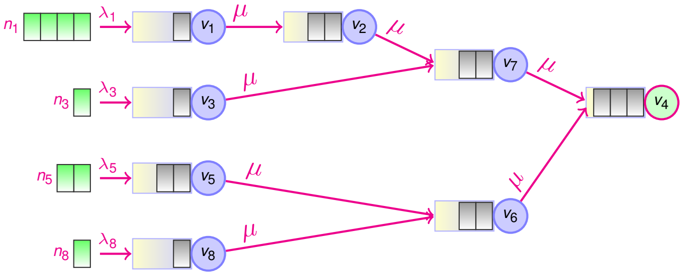
by setting: $\lambda_i = \frac{\mu n_i}{2n}$, we obtain: $\rho_i < 1$

according to Jackson, stationary distribution exists

we add **dummy customers**

real customers see now the stationary state

Proof Overview – Applying Jackson's Theorem

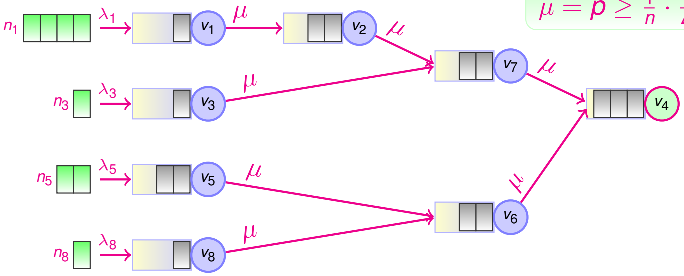


- we take all customers out of the system
- every customer will traverse through additional queues
- by setting: $\lambda_i = \frac{\mu n_i}{2n}$, we obtain: $\rho_i < 1$
- according to Jackson, stationary distribution exists

- we add **dummy customers**
- **real customers** see now the stationary state
- time needed to cross one queue is $\sim$ Exp()

Proof Overview – Applying Jackson's Theorem

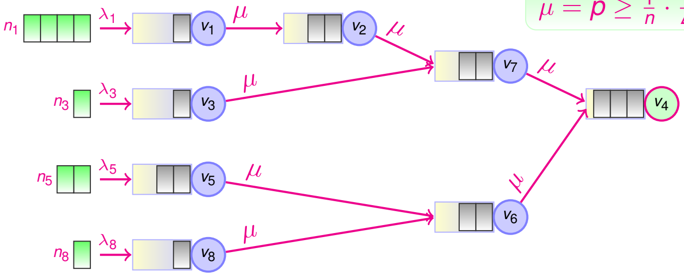
$\mu = p \geq \frac{1}{n} \cdot \frac{1}{\Delta}(1 - \frac{1}{q})$



- we take all customers out of the system
- every customer will traverse through additional queues
- by setting: $\lambda_i = \frac{\mu n_i}{2n}$, we obtain: $\rho_i < 1$
- according to Jackson, stationary distribution exists

- we add **dummy customers**
- **real customers** see now the stationary state
- time needed to cross one queue is $\sim$ Exp()
- v_4 finishes after $O(\Delta n)$ rounds

Proof Overview – Applying Jackson's Theorem

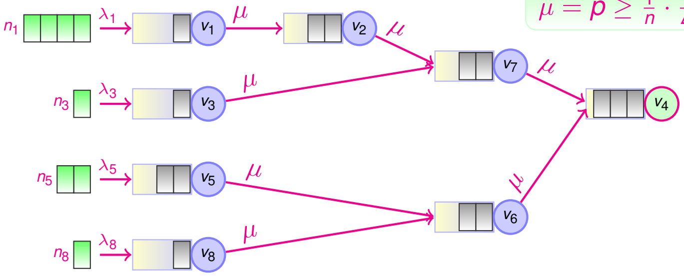


$\mu = p \geq \frac{1}{n} \cdot \frac{1}{\Delta}(1 - \frac{1}{q})$

v_4 finishes after $O(\Delta n)$ rounds with exponential high probability

Proof Overview – Applying Jackson's Theorem

$\mu = p \geq \frac{1}{n} \cdot \frac{1}{\Delta}(1 - \frac{1}{q})$



v_4 finishes after $O(\Delta n)$ rounds with exponential high probability

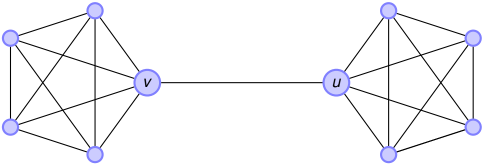
Using a union bound we obtain that the stopping time of all nodes is: $O(\Delta n)$ rounds

Proof Overview – Tightness of the Bound

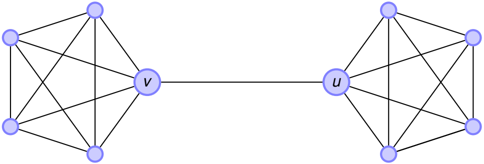
Corollary 3

There exists a graph with n nodes, for which stopping time of algebraic gossip is $\Omega(n^2)$.

Proof Overview – Worst Case Lower Bound

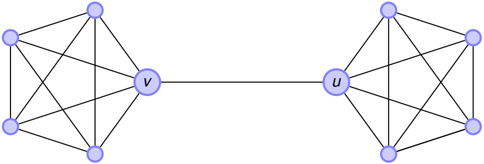


Proof Overview – Worst Case Lower Bound



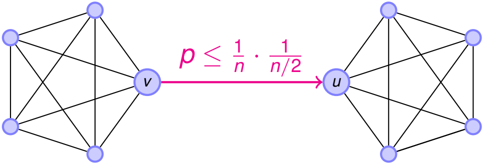
- Consider information flow form v to u.

Proof Overview – Worst Case Lower Bound



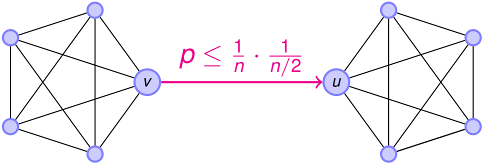
- Consider information flow form v to u.
- Only after receiving $\frac{n}{2}$ helpful messages, u finishes.

Proof Overview – Worst Case Lower Bound



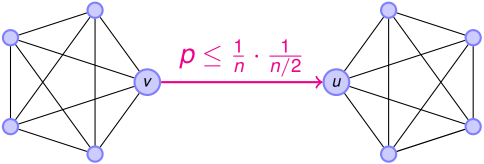
- Consider information flow form v to u.
- Only after receiving $\frac{n}{2}$ helpful messages, u finishes.
- At a given timeslot, helpful message from v to u is sent w.p. $p \leq \frac{1}{n} \cdot \frac{1}{n/2} = \frac{2}{n^2}$.

Proof Overview – Worst Case Lower Bound



- Consider information flow form v to u.
- Only after receiving $\frac{n}{2}$ helpful messages, u finishes.
- At a given timeslot, helpful message from v to u is sent w.p. $p \le \frac{1}{n} \cdot \frac{1}{n/2} = \frac{2}{n^2}$.
- Number of **timeslots** needed to u to receive $\frac{n}{2}$ helpful messages is the sum of $\frac{n}{2}$ geometric (p) r.v.

Proof Overview – Worst Case Lower Bound



- Consider information flow form v to u.
- Only after receiving $\frac{n}{2}$ helpful messages, u finishes.
- At a given timeslot, helpful message from v to u is sent w.p. $p \leq \frac{1}{n} \cdot \frac{1}{n/2} = \frac{2}{n^2}$.
- Number of **timeslots** needed to u to receive $\frac{n}{2}$ helpful messages is the sum of $\frac{n}{2}$ geometric (p) r.v.
- Using a Chernoff bound we obtain that u will finish after $\Omega(n^2)$ rounds.

Summary

- Stopping time of algebraic gossip on any graph is $O(\Delta n)$.

Summary

- Stopping time of algebraic gossip on any graph is $O(\Delta n)$.
- Worst case: $O(n^2)$, and there exists a graph for which it takes $\Omega(n^2)$.

Summary

- Stopping time of algebraic gossip on any graph is $O(\Delta n)$.
- Worst case: $O(n^2)$, and there exists a graph for which it takes $\Omega(n^2)$.
- For graphs with a constant maximum degree Δ, stopping time is: $O(n)$.

Summary

- Stopping time of algebraic gossip on any graph is $O(\Delta n)$.
- Worst case: $O(n^2)$, and there exists a graph for which it takes $\Omega(n^2)$.
- For graphs with a constant maximum degree Δ, stopping time is: $O(n)$.
- Thank you!

Deb, S., Médard, M., and Choute, C. (2006).
Algebraic gossip: a network coding approach to optimal multiple rumor mongering.
IEEE Transactions on Information Theory, 52(6):2486–2507.

Mosk-Aoyama, D. and Shah, D. (2006).
Information dissemination via network coding.
In *ISIT*, pages 1748–1752.