Broadcasting in an Unreliable SINR Model

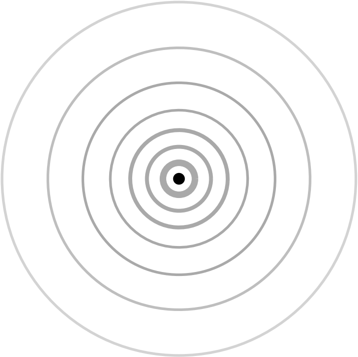
Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Philipp Schneider and Fabian Kuhn
University of Freiburg

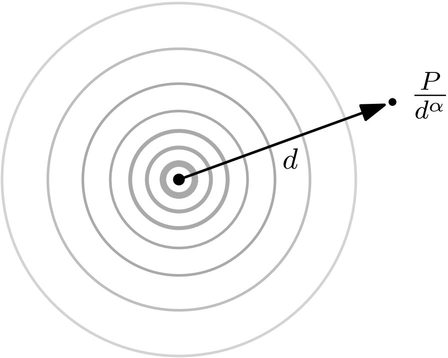
Signal Propagation in the SINR Model

Signal with transmission power P **fades with distance** from source.



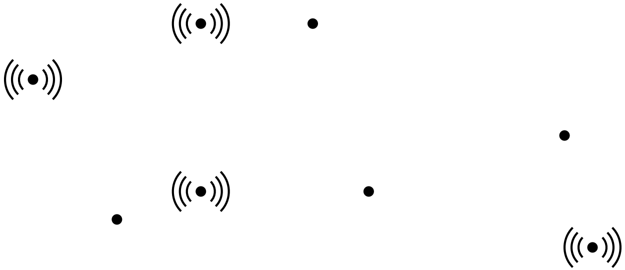
Signal Propagation in the SINR Model

Signal with transmission power P **fades with distance** from source.



Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.
Nodes *either send or listen*. Set of *interfering nodes* I.



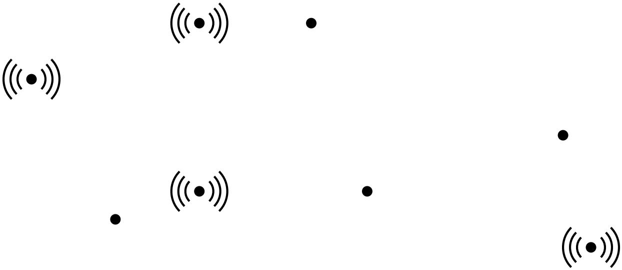
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w\in I} P/d(w,v)^{\alpha}} \geq \beta$$



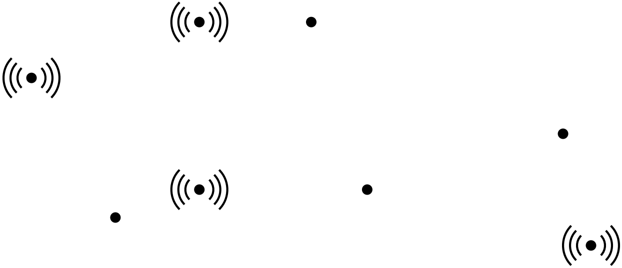
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power P*, *Noise N*, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N+\sum_{w\in I} P/d(w,v)^{\alpha}} \geq \beta$$



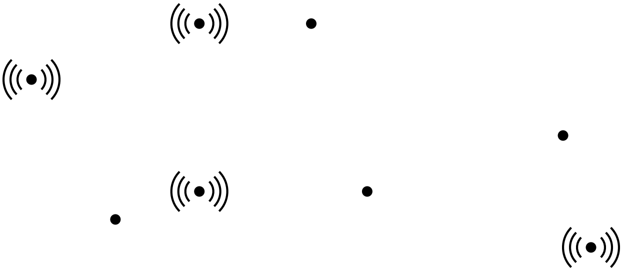
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta$$



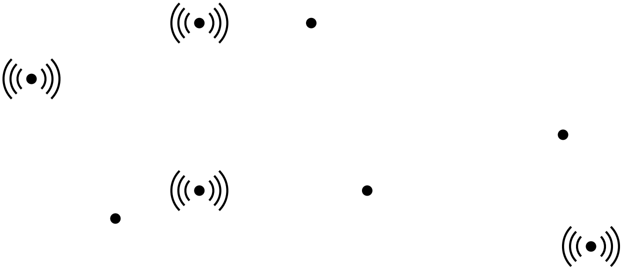
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta$$



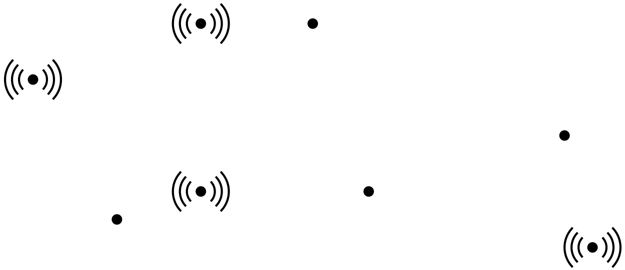
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power P*, *Noise N*, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta$$



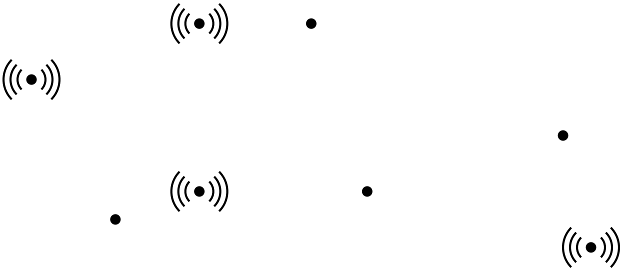
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta$$



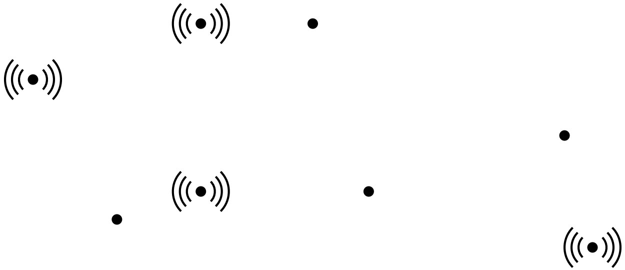
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w\in I} P/d(w,v)^{\alpha}} \geq \beta$$



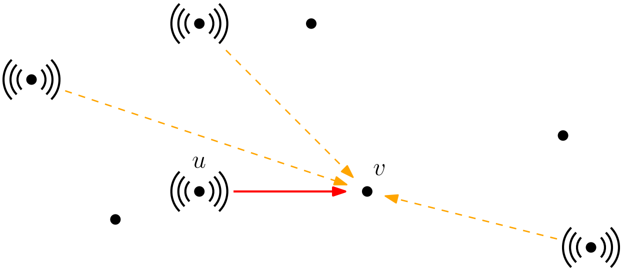
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N+\sum_{w\in I} P/d(w,v)^{\alpha}} < \beta$$



No Transmission!

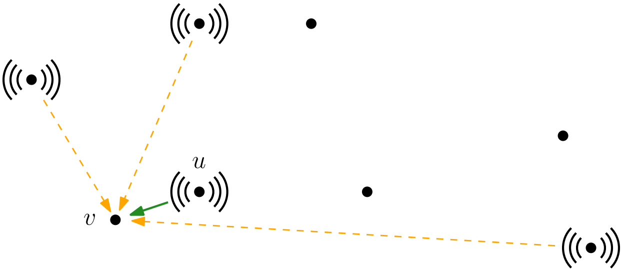
Signal to Interference and Noise Ratio - Model

Nodes V embedded to *metric space* (X,d). Time proceeds in *rounds*.

Nodes *either send or listen*. Set of *interfering nodes* I.

SINR parameters: *Power* P, *Noise* N, *Path loss exponent* α, *Threshold* β.

$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta$$



Transmission!

Modeling Choices

Minimal assumptions:

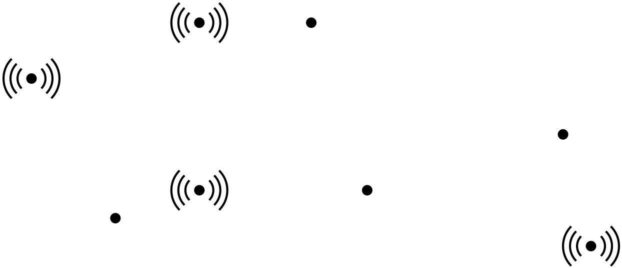
- No geometric information.
- No power control.
- No additional capabilities (e.g. carrier sensing).

Model Extension: Unreliable SINR Model

Silent node v receives transmission from sender u if and only if

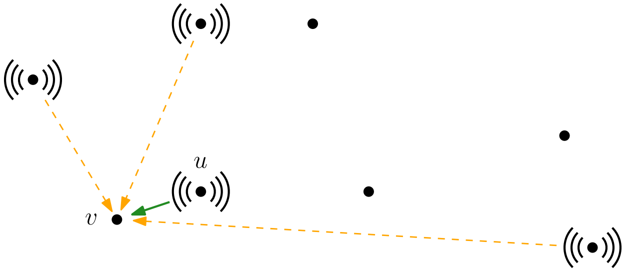
$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta_v$$

Adversary chooses $\beta_v \in [\beta_{\min}, \beta_{\max}]$ for each transmission.



Model Extension: Unreliable SINR Model

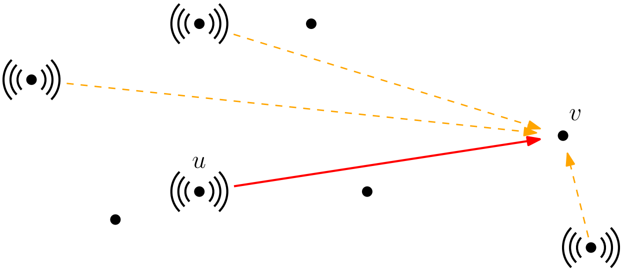
$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} \geq \beta_{\max}$$



Safe Transmission!

Model Extension: Unreliable SINR Model

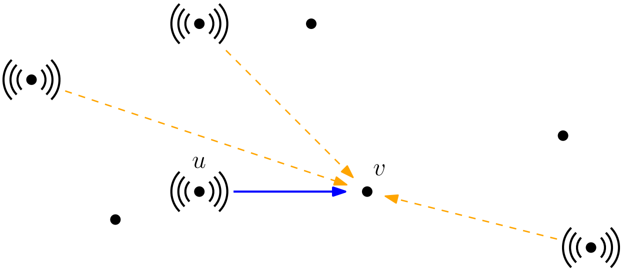
$$\mathrm{SINR}(u,v,I) := \frac{P/d(u,v)^{\alpha}}{N + \sum_{w \in I} P/d(w,v)^{\alpha}} < \beta_{\min}$$



No Transmission!

Model Extension: Unreliable SINR Model

$$\text{SINR}(u,v,I) := \frac{P/d(u,v)^\alpha}{N + \sum_{w \in I} P/d(w,v)^\alpha} \in [\beta_{\min}, \beta_{\max})$$



Adversary decides!

Adversary Motivation

In standard SINR models message reception is subject to a *deterministic* function.

- Real wireless transmission is *inherently unstable and unreliable*.
- Adversary adds a *dynamic* component.
- The proposed adversarial model captures a seemingly stronger adversary that manipulates *all* SINR-parameters (P, N, α, β).

The Broadcast Problem

Broadcast Problem

Broadcast is solved when message $\mathcal{M}$ is disseminated from a distinguished source node to all other nodes in V.

The Broadcast Problem

Broadcast Problem

Broadcast is solved when message $\mathcal{M}$ is disseminated from a distinguished source node to all other nodes in V.

Neighborhood Dissemination Problem

Neighborhood dissemination is solved when $\mathcal{M}$ is disseminated from $S \subseteq V$ to their Neighbors $N(S)$ in the *communication graph* G_C.

The Broadcast Problem

Broadcast Problem

Broadcast is solved when message $\mathcal{M}$ is disseminated from a distinguished source node to all other nodes in V.

Neighborhood Dissemination Problem

Neighborhood dissemination is solved when $\mathcal{M}$ is disseminated from $S \subseteq V$ to their Neighbors $N(S)$ in the *communication graph* G_C.

Communication graph

$G_C := (V, \{\{u,v\} \mid u,v \in V, u \neq v, d(u,v) \leq r_e\})$. Defined by the set of edges among nodes within *effective communication range* r_e of each other.

The Broadcast Problem

Broadcast Problem

Broadcast is solved when message $\mathcal{M}$ is disseminated from a distinguished source node to all other nodes in V.

Neighborhood Dissemination Problem

Neighborhood dissemination is solved when $\mathcal{M}$ is disseminated from $S \subseteq V$ to their Neighbors $N(S)$ in the *communication graph* G_C.

Communication graph

$G_C := (V, \{\{u,v\} \,|\, u,v \in V, u \neq v, d(u,v) \leq r_e\})$. Defined by the set of edges among nodes within *effective communication range* r_e of each other.

Observation: *Global Broadcast* can be solved by solving *Neighborhood Dissemination* D times where D is the *diameter* of G_C.

Solution Requirements

We want a *robust* randomized algorithm to solve broadcast, that works

- ... *with high probability* (w.h.p.),
  that is with probability at least $1 - \frac{1}{n^c}$, for constant c and $n := |V|$.
- ... for *any strategy* of the adversary.

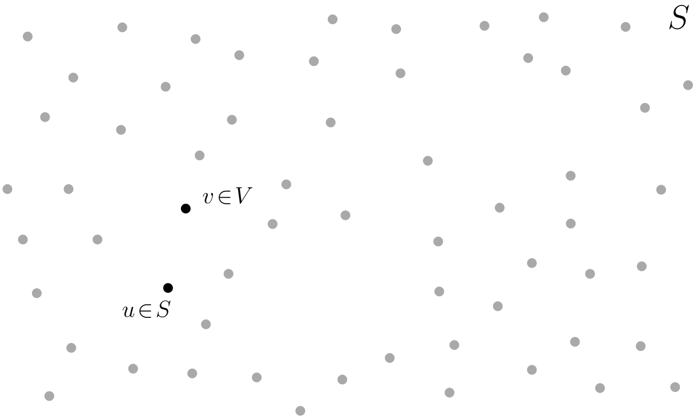
Previous Results and Our Result

Paper	Model	Broadcast
Daum et al., 2013	Reliable SINR	$\mathcal{O}\left(D \cdot \log n \cdot \log^{\star} n \cdot \operatorname{polylog} R\right)$
Jurdzinski et al., 2014	Reliable SINR	$\mathcal{O}(D \cdot \log^2 n)$
Halldórsson et al., 2015	Reliable SINR	$\mathcal{O}\left((D + \log n) \cdot \operatorname{polylog} R\right)$
This paper	Unreliable SINR	$\mathcal{O}\left(\frac{\beta_{\max}}{\beta_{\min}} \cdot D \cdot \log n \cdot \log^{\star} n \cdot \operatorname{polylog} R\right)$

R is the ***Ratio*** between the length of the *longest and shortest edge* in G_C.

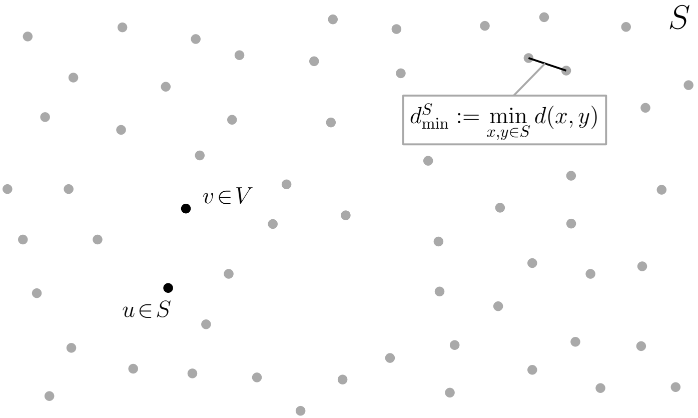
Communication Among Relatively Close Nodes

Lemma [cf. Daum et al., 2013]: Nodes $S \subseteq V$ send with probability p. If nodes $u \in S, v \in V$ are closer than a constant multiple of $d^S_{\min}$ and are in safe transmission range r_s of each other then a safe transmission from u to v takes place with *constant* probability $\mu \in (0, p)$.



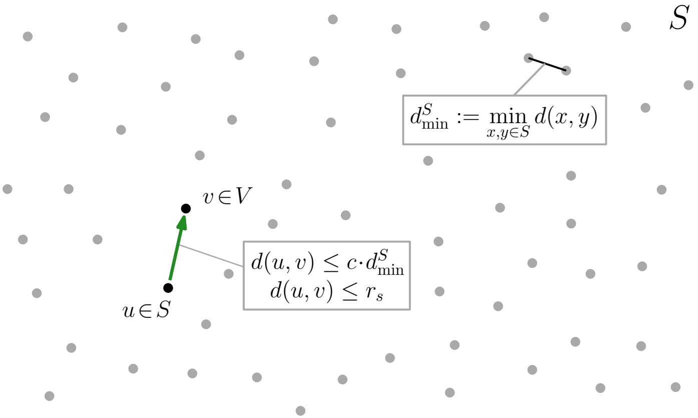
Communication Among Relatively Close Nodes

Lemma [cf. Daum et al., 2013]: Nodes $S \subseteq V$ send with probability p. If nodes $u \in S, v \in V$ are closer than a constant multiple of $d^S_{\min}$ and are in safe transmission range r_s of each other then a safe transmission from u to v takes place with *constant* probability $\mu \in (0, p)$.



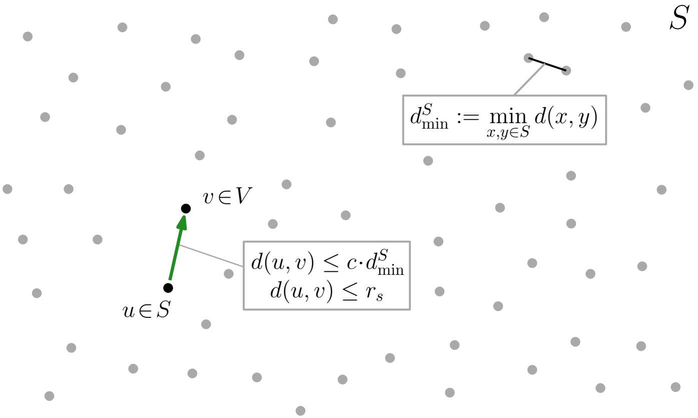
Communication Among Relatively Close Nodes

Lemma [cf. Daum et al., 2013]: Nodes $S \subseteq V$ send with probability p. If nodes $u \in S, v \in V$ are closer than a constant multiple of $d^S_{\min}$ and are in safe transmission range r_s of each other then a safe transmission from u to v takes place with *constant* probability $\mu \in (0, p)$.



Communication Among Relatively Close Nodes

Lemma [cf. Daum et al., 2013]: Nodes $S \subseteq V$ send with probability p. If nodes $u \in S, v \in V$ are closer than a constant multiple of $d^S_{\min}$ and are in safe transmission range r_s of each other then a safe transmission from u to v takes place with *constant* probability $\mu \in (0, p)$.



Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

- Algorithm proceeds in phases.

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.

Neighborhood Dissemination Algorithm

Algorithm RobustDissemination

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.
- **Property:** Nodes in $N(S)$ that did not yet receive $\mathcal{M}$ still have a close neighbor in S_ϕ.

Neighborhood Dissemination Algorithm

Algorithm RobustDissemination

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds ($S_1 := S$)
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ ($S_{\phi+1} \subseteq S_\phi$)

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.
- **Property:** Nodes in $N(S)$ that did not yet receive $\mathcal{M}$ still have a close neighbor in S_ϕ.
- *SINR-induced graph* $H[S_\phi]$ contains short edges among nodes S_ϕ.

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.
- **Property:** Nodes in $N(S)$ that did not yet receive $\mathcal{M}$ still have a close neighbor in S_ϕ.

- *SINR-induced graph* $H[S_\phi]$ contains short edges among nodes S_ϕ.
- Compute *Dominating Independet Set*.

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds ($S_1 := S$)
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ ($S_{\phi+1} \subseteq S_\phi$)

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.
- **Property:** Nodes in $N(S)$ that did not yet receive $\mathcal{M}$ still have a close neighbor in S_ϕ.
- *SINR-induced graph* $H[S_\phi]$ contains short edges among nodes S_ϕ.
- Compute *Dominating Independet Set*.
- $d_{\min}^{S_\phi}$ doubles each phase.

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.
- **Property:** Nodes in $N(S)$ that did not yet receive $\mathcal{M}$ still have a close neighbor in S_ϕ.

- *SINR-induced graph* $H[S_\phi]$ contains short edges among nodes S_ϕ.
- Compute *Dominating Independet Set*.
- $d_{\min}^{S_\phi}$ doubles each phase.
- In final phase ψ: $d_{\min}^{S_\psi}$ is large.

Neighborhood Dissemination Algorithm

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $(S_1 := S)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $(S_{\phi+1} \subseteq S_\phi)$

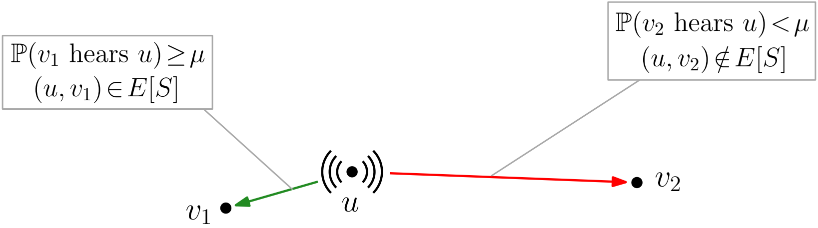
- Algorithm proceeds in phases.
- S_ϕ tries to disseminate $\mathcal{M}$.
- $S_{\phi+1} \subseteq S_\phi$ is 'thinned out'.
- **Property:** Nodes in $N(S)$ that did not yet receive $\mathcal{M}$ still have a close neighbor in S_ϕ.
- *SINR-induced graph* $H[S_\phi]$ contains short edges among nodes S_ϕ.
- Compute *Dominating Independet Set*.
- $d_{\min}^{S_\phi}$ doubles each phase.
- In final phase ψ: $d_{\min}^{S_\psi}$ is large.
- **Lemma**: Neighbors of S_ψ receive $\mathcal{M}$.

SINR-Induced Graphs - Without Adversary

$S \subseteq V$: sending with probability p.

$H[S]$ has **nodes** S and **reliable edges** $E[S]$.

$E[S]$ contains (u, v) iff **v receives message from u with probability** $\geq \mu$.

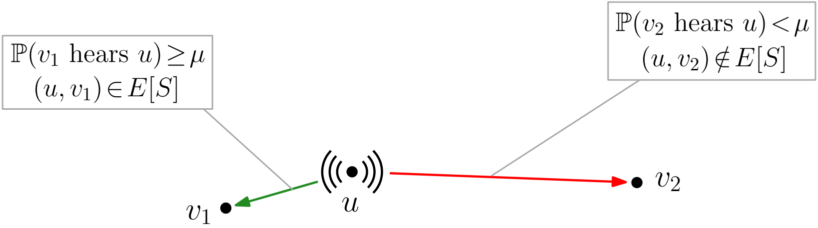


SINR-Induced Graphs - Without Adversary

$S \subseteq V$: sending with probability p.

$H[S]$ has **nodes** S and **reliable edges** $E[S]$.

$E[S]$ contains (u,v) iff **v receives message from u with probability** $\geq \mu$.



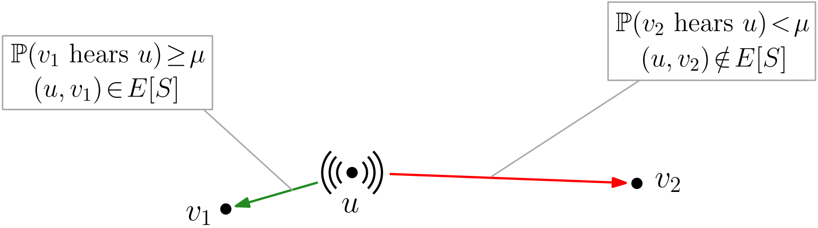
Previous Lemma: Choose μ such that *short edges* $(d(u,v) \leq 2d^S_{\min})$ are in $E[S]$.

SINR-Induced Graphs - Without Adversary

$S \subseteq V$: sending with probability p.

$H[S]$ has **nodes** S and **reliable edges** $E[S]$.

$E[S]$ contains (u,v) iff **v receives message from u with probability** $\geq \mu$.



Previous Lemma: Choose μ such that *short edges* ($d(u,v) \leq 2d^S_{\min}$) are in $E[S]$.

Constant degree of $\Delta \leq 1/\mu$.

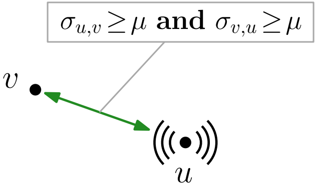
SINR-Induced Graphs - With Adversary

$$\sigma_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\max}\big) \quad \tau_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\min}\big).$$

SINR-Induced Graphs - With Adversary

$\sigma_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\max}\big) \quad \tau_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\min}\big).$

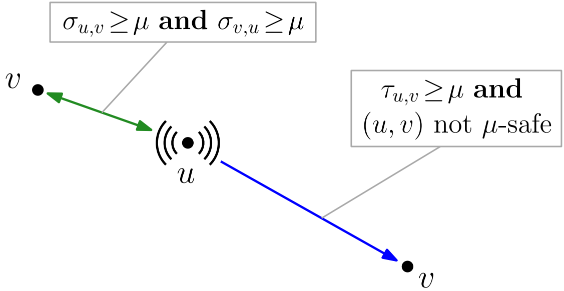
- $H[S]$ **contains all** μ-safe edges



SINR-Induced Graphs - With Adversary

$\sigma_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\max}\big) \quad \tau_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\min}\big).$

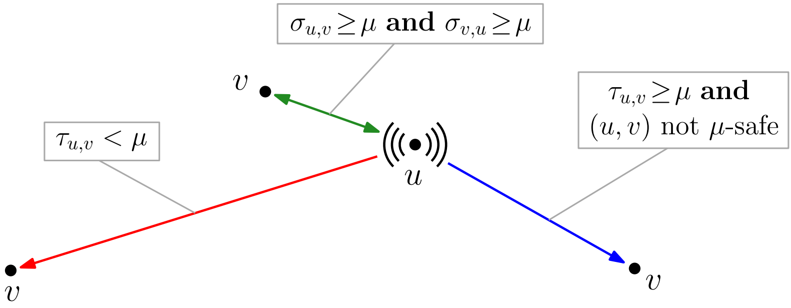
- $H[S]$ **contains all** μ-safe edges
- $H[S]$ **may contain** μ-unsafe edges (adversary decides)



SINR-Induced Graphs - With Adversary

$\sigma_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\max}\big) \quad \tau_{u,v} := \mathbb{P}\big(SINR(u,v,I) \geq \beta_{\min}\big).$

- $H[S]$ **contains all** μ-safe edges
- $H[S]$ **may contain** μ-unsafe edges (adversary decides)
- $H[S]$ **does not contain any** other edges



SINR-Induced Graphs - Computation

- $H[S]$ **cannot be pre-computed** due to the adversary.
- Instead provide **sub-procedure** TRANSMIT that nodes in S execute.
- For pair $u, v \in S$ participating in TRANSMIT:
  - If $\{u, v\}$ is μ-safe: Message transmitted.
  - If (u, v) is μ-unsafe: Adversary decides.
  - Otherwise: Message not transmitted.
- $\Rightarrow$ **Edges along which transmission takes place induce** $H[S]$.
- TRANSMIT **probes edges** by sending with prob. p for $\mathcal{O}(\log n)$ rounds.
- TRANSMIT **allows transmission** if sufficiently many probes were successful.

Dominating Independent Set

Dominating Independent Set (DIS) [Censor-Hillel et al., 2014]

Let $G = (V, E, E')$ be a graph with disjoint edge sets E and E'. A *DIS* $D \subseteq V$ of G is *independent* w.r.t. E and *dominating* w.r.t. $E \cup E'$.

Independent Set: Let $G = (V, E)$ be a graph. $Ind \subseteq V$ of G is *independent* if for all $u, v \in Ind$ there is **no** edge $\{u, v\} \in E$.

Dominating Set: Let $G = (V, E')$ be a graph. $Dom \subseteq V$ of G is *dominating* if for all $v \in V \setminus Dom$ there is a node $u \in Dom$ and an edge $(u, v) \in E'$.

Dominating Independent Set of $H[S]$

Edges $E[S]$ of $H[S]$ can be partitioned into

- $E_{\text{safe}}[S]$: μ-safe edges
- $E_{\text{unsafe}}[S]$: μ-unsafe edges .

Algorithms ComputeDIS($S, E_{\text{safe}}[S], E_{\text{unsafe}}[S]$)

---

1. Combine algorithm by [Linial, 1992] with Transmit:
   Obtain $\mathcal{O}(1)$-coloring w.r.t. $E_{\text{safe}}[S]$ in $\mathcal{O}(\log n \log^\star n)$ rounds.
2. For each color do
   3. Active nodes of current color join DIS ...
   4. ... and deactivate their neighbors via Transmit in $\mathcal{O}(\log n)$.

Dominating Independent Set of $H[S]$

Edges $E[S]$ of $H[S]$ can be partitioned into

- $E_{\text{safe}}[S]$: μ-safe edges
- $E_{\text{unsafe}}[S]$: μ-unsafe edges .

Algorithms ComputeDIS$(S, E_{\text{safe}}[S], E_{\text{unsafe}}[S])$

---

1. Combine algorithm by [Linial, 1992] with Transmit: Obtain $\mathcal{O}(1)$-coloring w.r.t. $E_{\text{safe}}[S]$ in $\mathcal{O}(\log n \log^{\star} n)$ rounds.
2. For each color do
   3. Active nodes of current color join DIS ...
   4. ... and deactivate their neighbors via Transmit in $\mathcal{O}(\log n)$.

Dominating Independent Set of $H[S]$

Edges $E[S]$ of $H[S]$ can be partitioned into

- $E_{\text{safe}}[S]$: μ-safe edges
- $E_{\text{unsafe}}[S]$: μ-unsafe edges .

Algorithms ComputeDIS$(S, E_{\text{safe}}[S], E_{\text{unsafe}}[S])$

---

1. Combine algorithm by [Linial, 1992] with Transmit:
   Obtain $\mathcal{O}(1)$-coloring w.r.t. $E_{\text{safe}}[S]$ in $\mathcal{O}(\log n \log^{\star} n)$ rounds.
2. For each color do
   3. Active nodes of current color join DIS ...
   4. ... and deactivate their neighbors via Transmit in $\mathcal{O}(\log n)$.

Dominating Independent Set of $H[S]$

Edges $E[S]$ of $H[S]$ can be partitioned into

- $E_{\text{safe}}[S]$: μ-safe edges
- $E_{\text{unsafe}}[S]$: μ-unsafe edges .

Algorithms ComputeDIS($S, E_{\text{safe}}[S], E_{\text{unsafe}}[S]$)

---

1 Combine algorithm by [Linial, 1992] with Transmit:
Obtain $\mathcal{O}(1)$-coloring w.r.t. $E_{\text{safe}}[S]$ in $\mathcal{O}(\log n \log^{\star} n)$ rounds.

2 For each color do

3 Active nodes of current color join DIS ...

4 ... and deactivate their neighbors via Transmit in $\mathcal{O}(\log n)$.

Dominating Independent Set of $H[S]$

Edges $E[S]$ of $H[S]$ can be partitioned into

- $E_{\text{safe}}[S]$: μ-safe edges
- $E_{\text{unsafe}}[S]$: μ-unsafe edges .

Algorithms ComputeDIS$(S, E_{\text{safe}}[S], E_{\text{unsafe}}[S])$

---

1. Combine algorithm by [Linial, 1992] with Transmit:
   Obtain $\mathcal{O}(1)$-coloring w.r.t. $E_{\text{safe}}[S]$ in $\mathcal{O}(\log n \log^{\star} n)$ rounds.
2. For each color do
   3. Active nodes of current color join DIS ...
   4. ... and deactivate their neighbors via Transmit in $\mathcal{O}(\log n)$.

Neighbors w.r.t. $E_{\text{safe}}[S]$ are differently colored $\Rightarrow$ **Independence w.r.t. $E_{\text{safe}}[S]$**.

Dominating Independent Set of $H[S]$

Edges $E[S]$ of $H[S]$ can be partitioned into

- $E_{\text{safe}}[S]$: μ-safe edges
- $E_{\text{unsafe}}[S]$: μ-unsafe edges .

Algorithms ComputeDIS$(S, E_{\text{safe}}[S], E_{\text{unsafe}}[S])$

---

1. Combine algorithm by [Linial, 1992] with Transmit: Obtain $\mathcal{O}(1)$-coloring w.r.t. $E_{\text{safe}}[S]$ in $\mathcal{O}(\log n \log^{\star} n)$ rounds.
2. For each color do
   3. Active nodes of current color join DIS ...
   4. ... and deactivate their neighbors via Transmit in $\mathcal{O}(\log n)$.

Neighbors w.r.t. $E_{\text{safe}}[S]$ are differently colored $\Rightarrow$ **Independence w.r.t. $E_{\text{safe}}[S]$**.

Node deactivated via Transmit $\Rightarrow$ **Node dominated w.r.t. $E_{\text{safe}}[S] \cup E_{\text{unsafe}}[S]$**.

Summary

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $\quad \mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $\quad \mathcal{O}\left(\text{polylog}\, n\right)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. $\quad Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

Summary

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds — $\mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ — $\mathcal{O}\left(\text{polylog}\, n\right)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. — $Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

Summary

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $\qquad \mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $\qquad \mathcal{O}\left(\text{polylog}\, n\right)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. $\qquad Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

Summary

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $\quad \mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $\quad \mathcal{O}\left(\text{polylog}\, n\right)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. $\quad Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

Summary

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q\log n)$ rounds — $\mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ — $\mathcal{O}\left(\text{polylog}\, n\right)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. $Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

Summary

Algorithm RobustDissemination

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $\quad \mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $\quad \mathcal{O}\left(\text{polylog}\, n\right)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. $\quad Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

$\Rightarrow \mathcal{O}\left(\text{polylog}(n+R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$ rounds to solve neighborhood dissemination.

Summary

Algorithm ROBUSTDISSEMINATION

For phase $\phi = 1$ to $\Theta(\log R)$ do:

1. S_ϕ sends with prob. $\frac{p}{Q}$ for $\mathcal{O}(Q \log n)$ rounds $\quad \mathcal{O}\left(\log n \cdot \text{polylog}(R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$
2. Determine DIS $S_{\phi+1}$ of $H[S_\phi]$ $\quad \mathcal{O}(\text{polylog}\, n)$

Property 1: There exists Q such that nodes in $N(S)$ that did not receive $\mathcal{M}$ yet, still have a neighbor in S_ϕ. $\quad Q \in \mathcal{O}\left(\text{polylog}\, R \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$

Property 2: In final phase ψ remaining active nodes S_ψ are 'sparse'
$\Rightarrow$ all neighbors of S_ψ receive $\mathcal{M}$ (Lemma).

$\Rightarrow \mathcal{O}\left(\text{polylog}(n+R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$ rounds to solve neighborhood dissemination.
$\Rightarrow \mathcal{O}\left(D \cdot \text{polylog}(n+R) \cdot \frac{\beta_{\max}}{\beta_{\min}}\right)$ rounds to solve broadcast in the **Unreliable Model**.

Thank you.

References I

- Censor-Hillel, K. et al. (2014). “Structuring unreliable radio networks”. In: *Distributed Computing* 27.1.
- Daum, S. et al. (2013). “Broadcast in the Ad Hoc SINR Model”. In: *Proceedings of the 27th International Symposium on Distributed Computing*. DISC '13. Jerusalem, Israel: Springer-Verlag.
- Erdös, P., P. Frankl und Z. Füredi (1985). “Families of finite sets in which no set is covered by the union of r others”. In: *Israel Journal of Mathematics* 51.1.
- Halldórsson, M. M., S. Holzer und N. Lynch (2015). “A Local Broadcast Layer for the SINR Network Model”. In: *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. PODC '15. Donostia-San Sebastián, Spain: ACM.

References II

Jurdzinski, T. et al. (2014). "On the impact of geometry on ad hoc communication in wireless networks". In: *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*. PODC '14. ACM.

Linial, N. (1992). "Locality in Distributed Graph Algorithms". In: *SIAM J. Comput.* 21.1.