Fast Computing in Networks with Limited Bandwidth

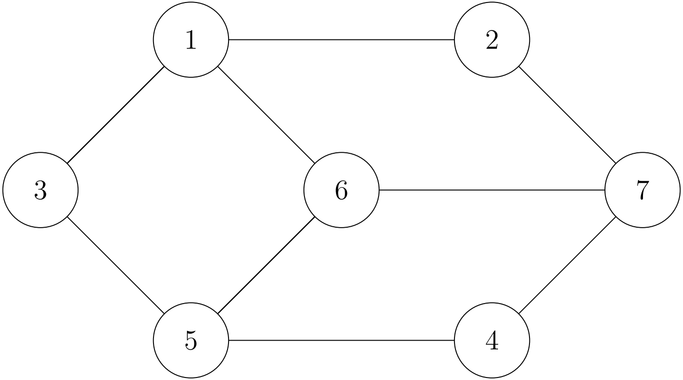
Ph.D Candidate:
Dennis **Olivetti**

Supervisor:
Prof. Pierre **Fraigniaud**

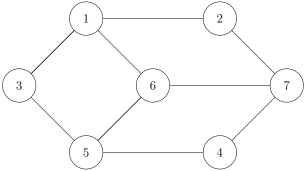
Outline

- CONGEST Model
- Subgraph Detection
- Core-Periphery Networks
- Bandwidth Tradeoffs

Distributed Computing

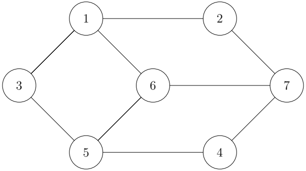


CONGEST model



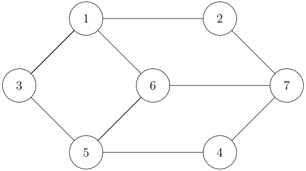
- Synchronous
  - All nodes start the computation at the same round
  - The computation proceeds in phases
  - At each phase each node can send a different message to each neighbor

CONGEST model



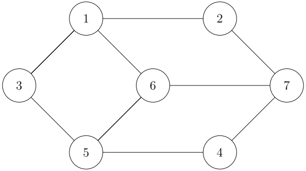
- Synchronous
  - All nodes start the computation at the same round
  - The computation proceeds in phases
  - At each phase each node can send a different message to each neighbor
- Fault-free

CONGEST model



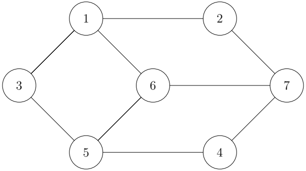
- Synchronous
  - All nodes start the computation at the same round
  - The computation proceeds in phases
  - At each phase each node can send a different message to each neighbor
- Fault-free
- Non-rational agents

CONGEST model



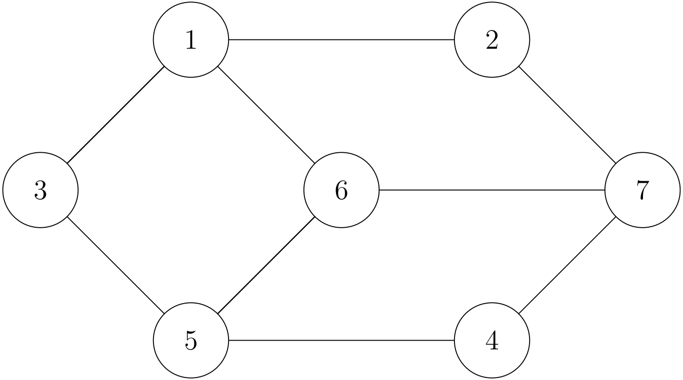
- Synchronous
  - All nodes start the computation at the same round
  - The computation proceeds in phases
  - At each phase each node can send a different message to each neighbor
- Fault-free
- Non-rational agents
- Limited bandwidth

CONGEST model

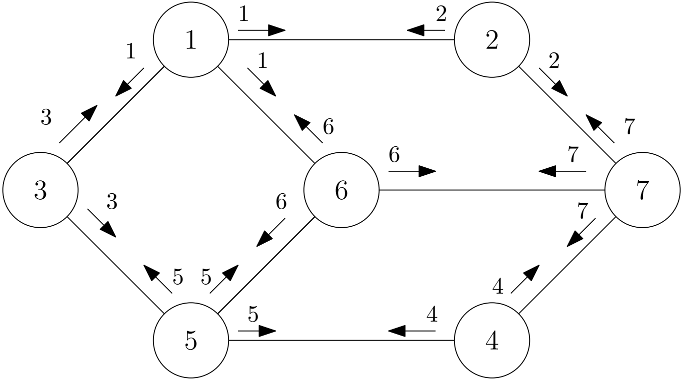


- Synchronous
  - All nodes start the computation at the same round
  - The computation proceeds in phases
  - At each phase each node can send a different message to each neighbor
- Fault-free
- Non-rational agents
- Limited bandwidth
- Complexity:
  - Number of rounds
  - Number of messages

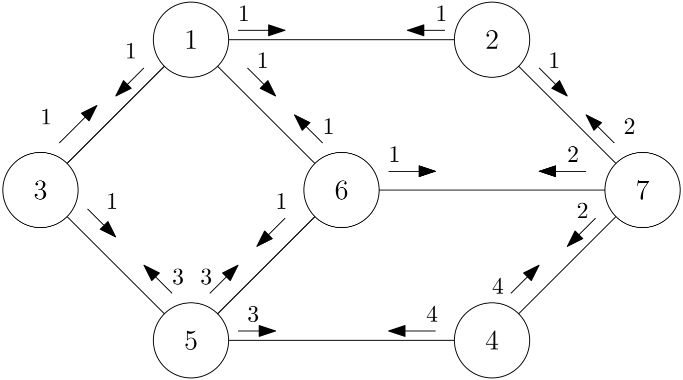
Example: Finding a leader



Example: Finding a leader



Example: Finding a leader



Example: Finding a leader

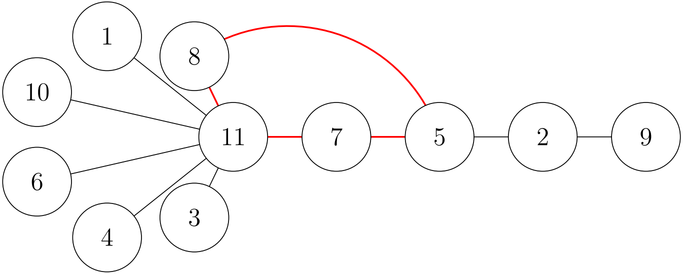
- Time complexity: $\Theta(diameter)$
- Message complexity: $\Theta(edges \cdot diameter)$

Complexities

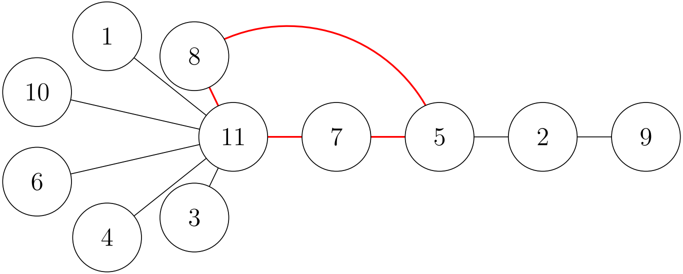
Complexity	Problem
$\Theta(1)$	Checking a coloring
$\Theta(\log^* n)$[1,2]	Coloring a ring
$O(\log n)$[3]	Approximating a maximal matching
$\widetilde{\Theta}(\sqrt{n} + \mathit{diameter})$[4,5]	MST, SSSP approximation
$\Theta(n / \log n)$[6]	Diameter, APSP
$\widetilde{O}(n^{5/4})$[7]	Weighted APSP
$\Omega(n^2 / \log^2 n)$[8]	Maximum independent set
$\Theta(n^2)$[8]	Checking a non trivial automorphism

[1][Linial '92] [2][Cole, Vishkin '86] [3][Israeli, Itai '86] [4][Kutten, Peleg '98]
[5][Becker, Karrenbauer, Krinninger, Lenzen '16]
[6][Hua, Fan, Qian, Ai, Li, Shi, Jin '16] [7][Huang, Nanongkai, Saranurak '17]
[8][Censor-Hillel, Khoury, Paz '17]

Example: Congestion



Example: Congestion

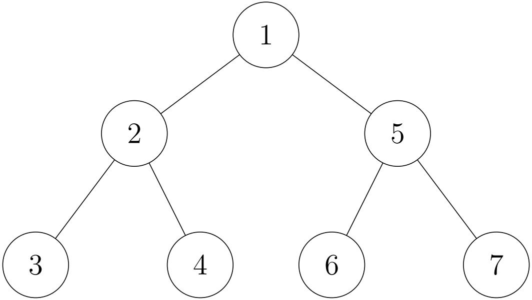


C_4-detection requires $\widetilde{\Theta}(\sqrt{n})$. [Drucker, Kuhn, Oshman, PODC'14]

Subgraph detection

Given a graph pattern H:

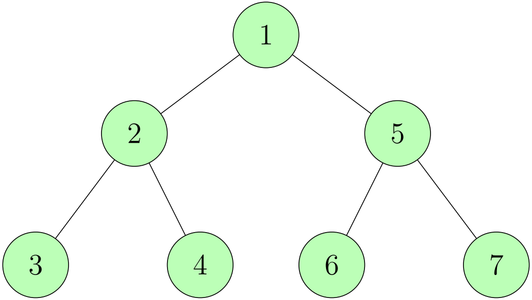
- if G does not contain H as subgraph, all nodes accept;
- otherwise, at least one node rejects.



Subgraph detection

Given a graph pattern H:

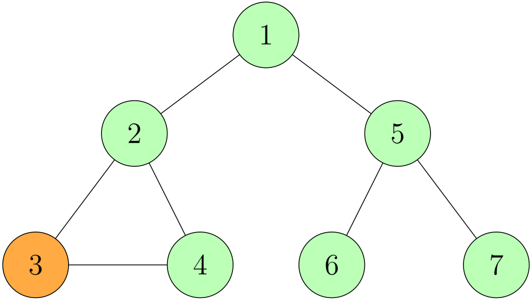
- if G does not contain H as subgraph, all nodes accept;
- otherwise, at least one node rejects.



Subgraph detection

Given a graph pattern H:

- if G does not contain H as subgraph, all nodes accept;
- otherwise, at least one node rejects.



Subgraph detection

Results:

- Triangle detection requires $O(n^{2/3})$ rounds [Izumi, Le Gall, PODC'17].

Subgraph detection

Results:

- Triangle detection requires $O(n^{2/3})$ rounds [Izumi, Le Gall, PODC'17].
- K_k detection ($k \geq 4$) requires $\widetilde{\Omega}(n)$ rounds. [Drucker, Kuhn, Oshman, PODC'14]

Subgraph detection

Results:

- Triangle detection requires $O(n^{2/3})$ rounds [Izumi, Le Gall, PODC'17].
- K_k detection ($k \geq 4$) requires $\widetilde{\Omega}(n)$ rounds. [Drucker, Kuhn, Oshman, PODC'14]
- C_k detection ($k \geq 4$) requires $\widetilde{\Omega}(ex(n, C_k)/n)$ rounds, where $ex(n, H)$ is the maximal possible number of edges of an n-node graph G such that G does not contain H as subgraph. [Drucker, Kuhn, Oshman, PODC'14]

Subgraph detection

Results:

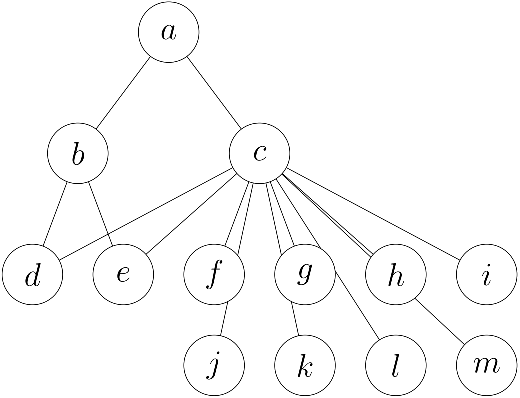
- Triangle detection requires $O(n^{2/3})$ rounds [Izumi, Le Gall, PODC'17].
- K_k detection ($k \geq 4$) requires $\widetilde{\Omega}(n)$ rounds. [Drucker, Kuhn, Oshman, PODC'14]
- C_k detection ($k \geq 4$) requires $\widetilde{\Omega}(ex(n, C_k)/n)$ rounds, where $ex(n, H)$ is the maximal possible number of edges of an n-node graph G such that G does not contain H as subgraph. [Drucker, Kuhn, Oshman, PODC'14]
- For every k-node graph H, H-detection requires $\widetilde{O}(n^{1-2/k})$ rounds, if the communication graph is a clique. [Dolev, Lenzen, Peled, DISC'12]

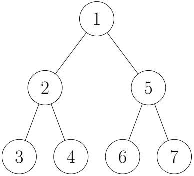
Which patterns are detectable efficiently?

Theorem [Fraigniaud, Montealegre, O., Rapaport, Todinca, DISC'17]

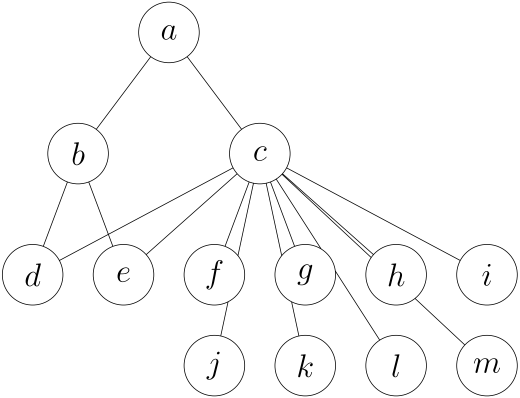
For every tree T of constant size, there exists a deterministic algorithm performing in $O(1)$ rounds in the CONGEST model for detecting whether the given input network contains T as a subgraph.

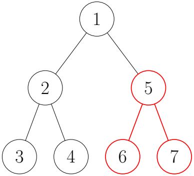
Tree Detection



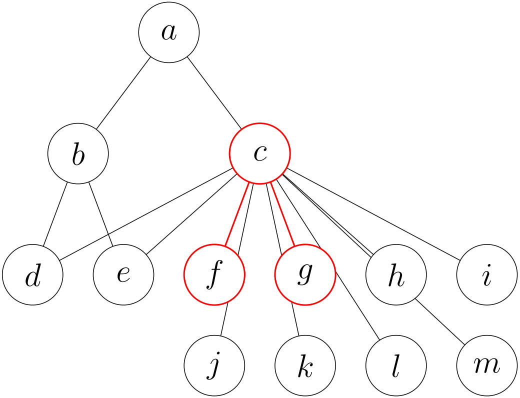


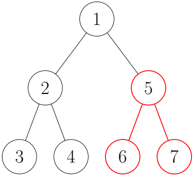
Tree Detection



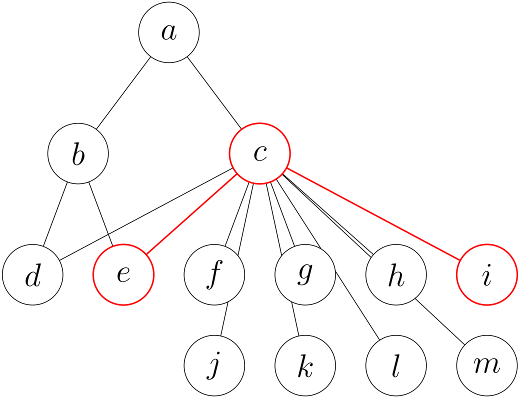


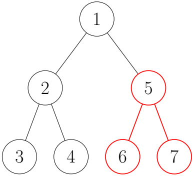
Tree Detection



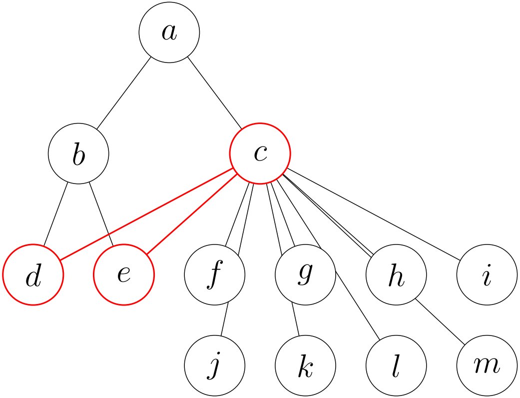


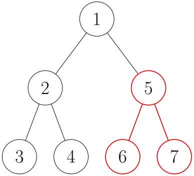
Tree Detection



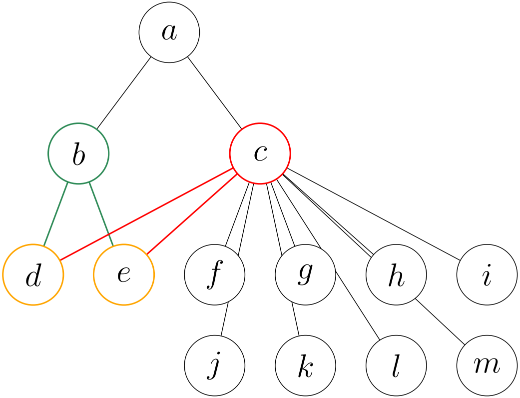


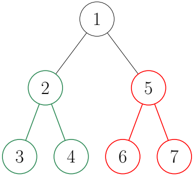
Tree Detection



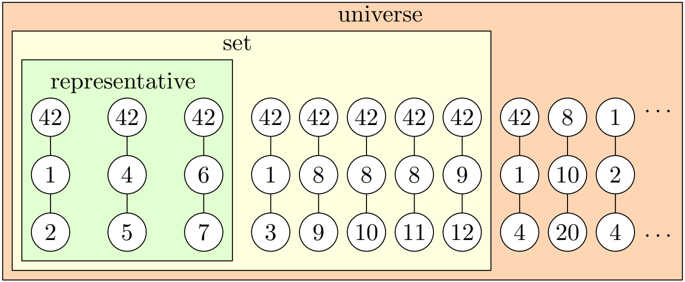


Tree Detection

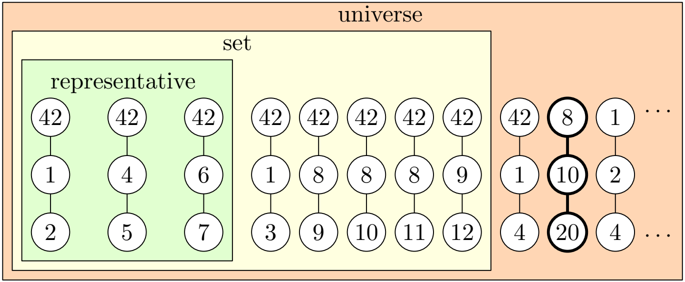




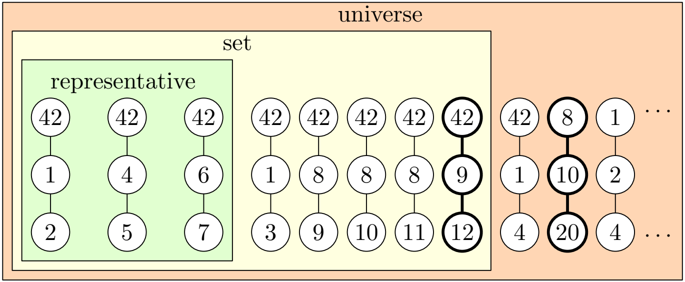
Representative Sets



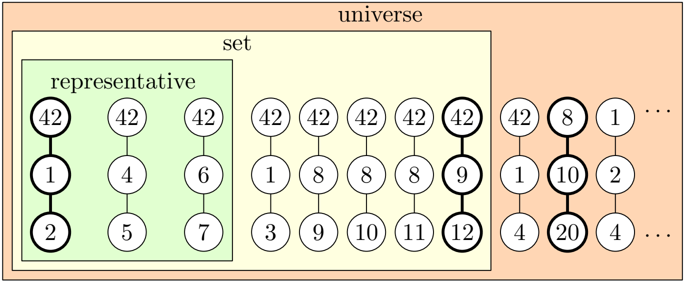
Representative Sets



Representative Sets



Representative Sets



Representative Sets

Lemma [Erdős, Hajnal, Moon '64]

Let V be a set of size n, and consider two integer parameters p and q. For any set $F \subseteq \mathcal{P}(V)$ of subsets of size at most p of V, there exists a *compact* (p, q)*-representation* of F, i.e., a subset $\hat{F}$ of F satisfying:

1. For each set $C \subseteq V$ of size at most q, if there is a set $L \in F$ such that $L \cap C = \emptyset$, then there also exists $\hat{L} \in \hat{F}$ such that $\hat{L} \cap C = \emptyset$;
2. The cardinality of $\hat{F}$ is at most $\binom{p+q}{p}$, for any $n \geq p + q$.

Breaking the lower bounds

Pattern detection:

- We want to detect more patterns, efficiently.
- There are polynomial lower bounds.
- We need to relax the problem.

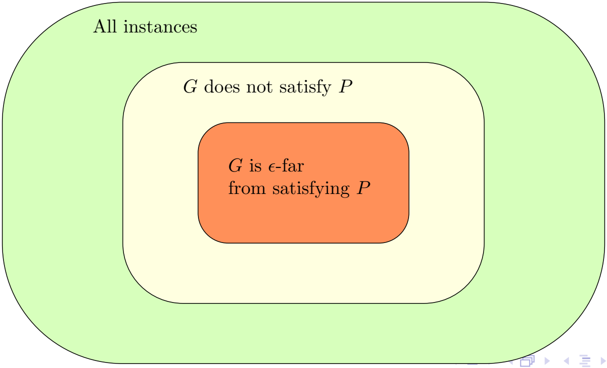
Possible ways:

- Allow more bandwidth.
- Assume that the communication graph is a clique.
- Allow some error.

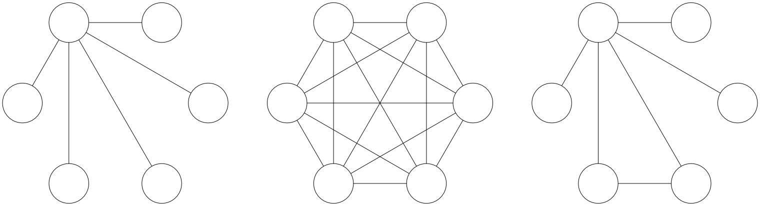
Distributed property testing

Let $G = (V, E)$, $n = |V|$, $m = |E|$. Let ϵ be a small constant in $(0, 1)$. A distributed tester for a graph property P is a distributed randomized algorithm A that satisfies the following conditions:

- G satisfies $P \Rightarrow$ every node outputs "accept"
- G is ϵ-far from satisfying $P \Rightarrow$ $\Pr[\text{at least one node outputs "reject"}] \geq \frac{2}{3}$



Distributed property testing



How to measure how far is a graph from satisfying a property?

There exist two distinct models:

How to measure how far is a graph from satisfying a property?

There exist two distinct models:

Dense model

A graph is ϵ-far from satisfying a property if at least ϵn^2 edges should be added or removed from G in order to make the property hold.

How to measure how far is a graph from satisfying a property?

There exist two distinct models:

Dense model

A graph is ϵ-far from satisfying a property if at least ϵn^2 edges should be added or removed from G in order to make the property hold.

Sparse model

A graph is ϵ-far from satisfying a property if at least ϵm edges should be added or removed from G in order to make the property hold.

State of the art

[Censor-Hillel, Fischer, Schwartzman, Vasudev '16]

Any ϵ-tester for the dense model (for a non-disjointed property) that makes q queries can be converted to a distributed ϵ-tester that requires $O(q^2)$ rounds in the distributed setting.

State of the art

[Censor-Hillel, Fischer, Schwartzman, Vasudev ’16]

Any ϵ-tester for the dense model (for a non-disjointed property) that makes q queries can be converted to a distributed ϵ-tester that requires $O(q^2)$ rounds in the distributed setting.

[Censor-Hillel, Fischer, Schwartzman, Vasudev ’16]

- Triangle freeness can be tested in $O(1/\epsilon^2)$
- Cycle freeness can be tested $O(\log n/\epsilon)$
- Cycle freeness requires at least $\Omega(\log n)$
- Bipartiteness can be tested in in $O(poly(\log \frac{n}{\epsilon}/\epsilon))$ in bounded degree graphs

State of the art

[Censor-Hillel, Fischer, Schwartzman, Vasudev '16]

Any ϵ-tester for the dense model (for a non-disjointed property) that makes q queries can be converted to a distributed ϵ-tester that requires $O(q^2)$ rounds in the distributed setting.

[Censor-Hillel, Fischer, Schwartzman, Vasudev '16]

- Triangle freeness can be tested in $O(1/\epsilon^2)$
- Cycle freeness can be tested $O(\log n/\epsilon)$
- Cycle freeness requires at least $\Omega(\log n)$
- Bipartiteness can be tested in in $O(poly(\log \frac{n}{\epsilon}/\epsilon))$ in bounded degree graphs

[Fraigniaud, Rapaport, Salo, Todinca '16]

- H-freeness can be tested in constant time for any H s.t. $|V(H)| \leq 4$

Which patterns are detectable efficiently?

Theorem [Fraigniaud, O., SPAA'17]

There exists an ϵ-tester for C_k freeness, for any constant $k \geq 3$, that requires $O(\frac{1}{\epsilon})$ rounds in the CONGEST model.

*[Three Notes on Distributed Property Testing, Even, Fischer, Fraigniaud, Gonen, Levi, Medina, Montealegre, O., Oshman, Rapaport, Todinca]

Which patterns are detectable efficiently?

Theorem [Fraigniaud, O., SPAA'17]

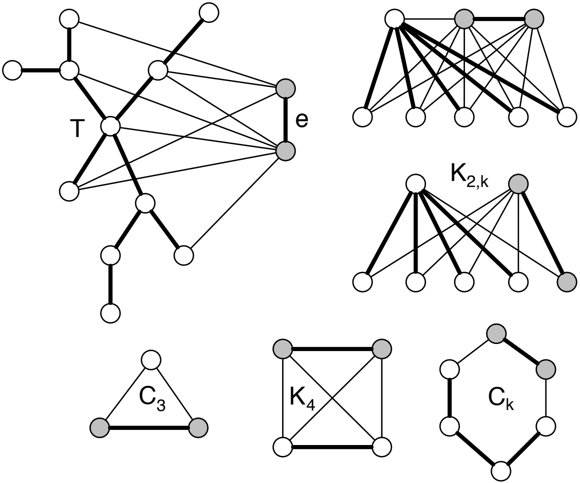
There exists an ϵ-tester for C_k freeness, for any constant $k \geq 3$, that requires $O(\frac{1}{\epsilon})$ rounds in the CONGEST model.

Theorem [Fraigniaud, Montealegre, O., Rapaport, Todinca, DISC'17]*

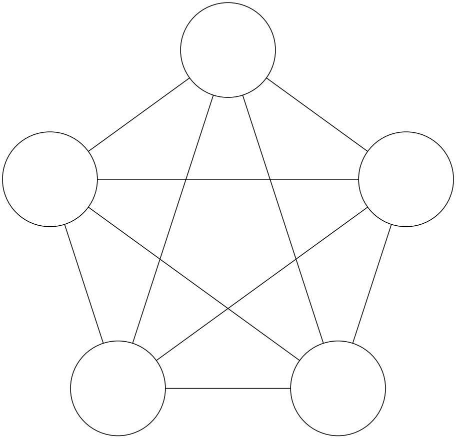
There exists an ϵ-tester for H freeness, for any graph H of constant size composed by a tree, an edge, and arbitrary connections between the endpoints of the edge and the nodes of the tree, that requires $O(\frac{1}{\epsilon})$ rounds in the CONGEST model.

*[Three Notes on Distributed Property Testing, Even, Fischer, Fraigniaud, Gonen, Levi, Medina, Montealegre, O., Oshman, Rapaport, Todinca]

Tree + 1 edge



Open Problem



Congested clique

If we assume that the communication graph is a clique we can solve many problems very efficiently:

- C_4 detection in $O(1)$[1] rounds.
- MST in $O(1)$[2] rounds.
- Matrix multiplication, APSP approximation, triangle and 4-cycle counting, girth computing in $O(n^{0.158})$[2] rounds.

[1][Censor-Hillel, Kaski, Korhonen, Lenzen, Paz, Suomela '15]
[2][Jurdzinski, Nowicki '17] [3][Drucker, Kuhn, Oshman '17]

Congested clique

If we assume that the communication graph is a clique we can solve many problems very efficiently:

- C_4 detection in $O(1)$[1] rounds.
- MST in $O(1)$[2] rounds.
- Matrix multiplication, APSP approximation, triangle and 4-cycle counting, girth computing in $O(n^{0.158})$[2] rounds.

This model is very powerful:

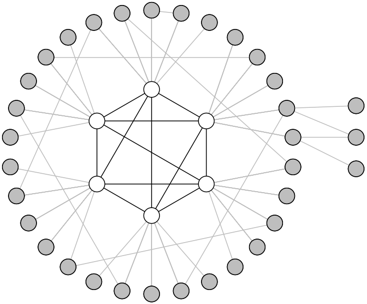
- No lower bounds are known.
- It can simulate some powerful classes of circuits.[3]

The number of edges is quadratic in the number of nodes, it may be hard to build it in practice.

[1] [Censor-Hillel, Kaski, Korhonen, Lenzen, Paz, Suomela '15]
[2] [Jurdzinski, Nowicki '17] [3] [Drucker, Kuhn, Oshman '17]

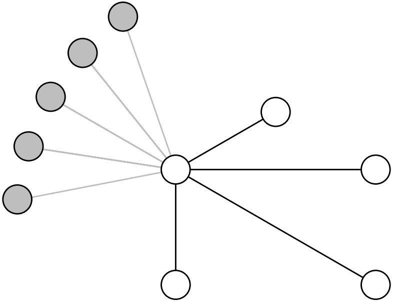
Core-periphery networks

- A novel network architecture for parallel and distributed computing, inspired by social networks and complex systems, proposed by [Avin, Borokhovich, Lotker, Peleg '14].
- A core-periphery network $G = (V, E)$ has its node set partitioned into a *core* C and a *periphery* P, and satisfies the following axioms:
  - Core boundary
  - Clique emulation
  - Periphery-core convergecast



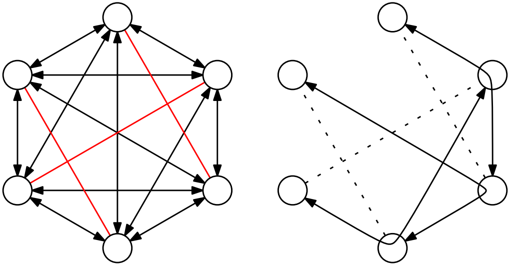
Axiom 1: Core boundary

For every node $v \in C$, $\deg_C(v) \simeq \deg_P(v)$, where, for $S \subseteq V$ and $v \in V$, $\deg_S(v)$ denotes the number of neighbors of v in S.



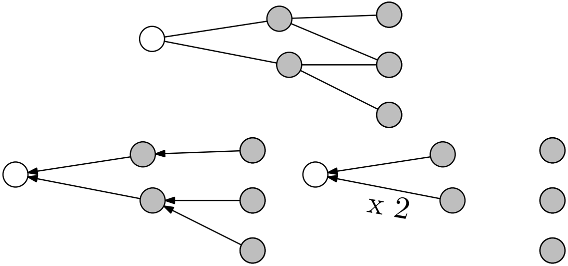
Axiom 2: Clique emulation

The core can emulate the clique in a constant number of rounds in the CONGEST model. That is, there is a communication protocol running in a constant number of rounds in the CONGEST model such that, assuming that each node $v \in C$ has a message $M_{v,w}$ on $O(\log n)$ bits for every $w \in C$, then, after $O(1)$ rounds, every $w \in C$ has received all messages $M_{v,w}$, for all $v \in C$.

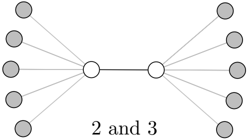


Axiom 3: Periphery-core convergecast

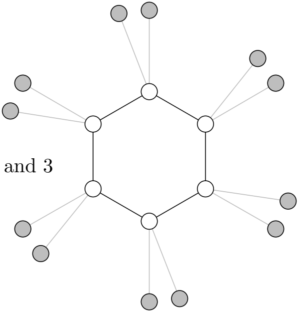
There is a communication protocol running in a constant number of rounds in the CONGEST model such that, assuming that each node $v \in P$ has a message M_v on $O(\log n)$ bits, then, after $O(1)$ rounds, for every $v \in P$, at least one node in the core has received M_v.



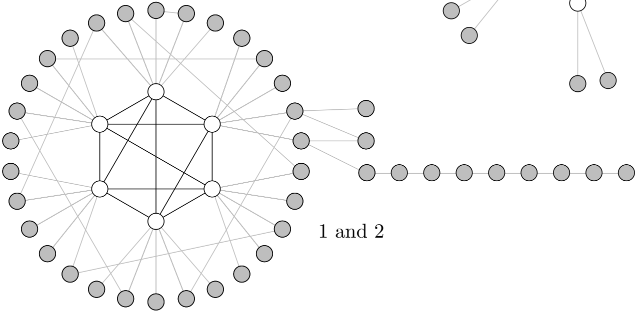
Not Core-Periphery Networks



2 and 3



1 and 3



1 and 2

Core-Periphery in Practice

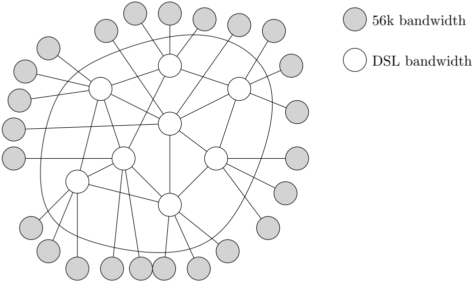
Many file sharing networks:

- Centralized:
  - eDonkey
  - OpenNap
- Distributed:
  - Kademlia
  - WinMX
  - Gnutella
    (Bearshare, Limewire, . . .)
  - Bittorrent

Core-Periphery in Practice

Many file sharing networks:

- Centralized:
  - eDonkey
  - OpenNap
- Distributed:
  - Kademlia
  - **WinMX**
  - **Gnutella**
    (Bearshare, Limewire, . . .)
  - Bittorrent



Which graphs can satisfy Axiom 2 efficiently?

- Nodes should be able to perform an all-to-all communication efficiently
- We want the graph as sparse as possible

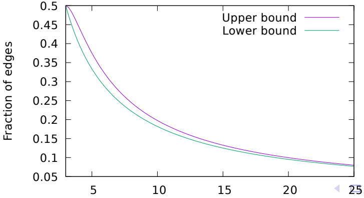
Tradeoff between edges and rounds

Theorem [Balliu, Fraigniaud, Lotker, O., SIROCCO'16]

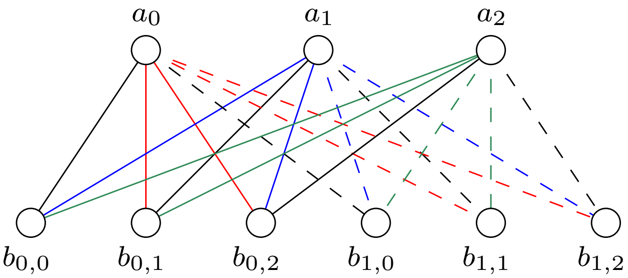
Let $n \geq 1$, and $k \geq 3$. There is an n-node graph with $\frac{k-2}{(k-1)^2}\, n^2$ edges that can emulate the n-node clique in k rounds. Also, there is an n-node graph with $\frac{1}{3}n^2$ edges that can emulate the n-node clique in 2 rounds.

Theorem [Balliu, Fraigniaud, Lotker, O., SIROCCO'16]

Let $n \geq 1$, $k \in \{1, \ldots, n-1\}$, and let G be an n-node graph that can emulate the n-node clique in k rounds. Then G has at least $\frac{n(n-1)}{k+1}$ edges.



Proof idea



Random graphs

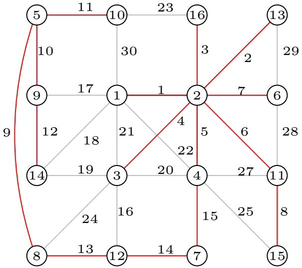
Theorem [Balliu,Fraigniaud, Lotker, O., SIROCCO'16]

Let $c \geq 0$, $n \geq 1$, $\alpha = \sqrt{(3+c)e/(e-2)}$ where e is the base of the natural logarithm, and $p \geq \alpha\sqrt{\ln n/n}$. For $G \in \mathcal{G}_{n,p}$,
$\Pr[G \text{ can emulate } K_n \text{ in } O(\min\{\frac{1}{p^2}, np\}) \text{ rounds}] \geq 1 - O(\frac{1}{n^{1+c}})$

The power of Core-Periphery networks

- Matrix transposition in $O(k)$ rounds, where k is the number of nonzero entries.
- Vector by matrix multiplication in $O(k)$ rounds.
- Matrix multiplication in $O(k^2)$ rounds.
- Rank finding in $O(1)$ rounds.
- Median finding in $O(1)$ rounds.
- Mode finding in $O(1)$ rounds.
- Number of distinct values $O(1)$ rounds.
- MST in $O(\log^2 n)$ rounds.

Minimum Spanning Tree



MST in the Congest model:

- $D = 1$:
  $O(1)$[1] randomized,
  $O(\log\log n)$[2] deterministic
- $D = 2$:
  $O(\log n)$[3] deterministic
- $D \geq 3$: $\Omega(\sqrt[3]{n})$[3]
- Core-Periphery ($D \approx 4$):
  $O(\log^2 n)$[4] randomized

[1][Jurdzinski, Nowicki '17] [2][Lotker, Patt-Shamir, Pavlov, Peleg '05]
[3][Lotker, Patt-Shamir, Peleg '06] [4][Avin, Borokhovich, Lotker, Peleg '14]

Minimum Spanning Tree

Theorem [Balliu,Fraigniaud, Lotker, O., SIROCCO'16]

There exists a deterministic algorithm that solves the MST construction task in $O(\log n)$ rounds in Core-Periphery networks.

Open problem

In the Congested Clique we can construct a MST in $O(1)$ rounds, can we construct a MST in Core-Periphery networks in $o(\log n)$?

The CONGEST$_B$ model

Typically, messages are chosen to be $B = O(\log n)$ bits:

- MST can be constructed in $O(\sqrt{n} \log^* n + D)$[1] rounds
- SSSP can be approximated in $\widetilde{O}(\epsilon^{-O(1)}(\sqrt{n} + D))$[2]
- APSP can be computed in $O(n/\log n)$[3]

Typically, lower bounds depend on B:

- MST and SSSP require $\Omega(\sqrt{n/B})$[4]
- APSP requires $\Omega(n/B)$[5,6]

[1] [Kutten, Peleg '98]

[2] [Becker, Karrenbauer, Krinninger, Lenzen '16]

[3] [Hua, Fan, Qian, Ai, Li, Shi, Jin '16]

[4] [Das Sarma, Holzer, Kor, Korman, Nanongkai, Pandurangan, Peleg, Wattenhofer '10]

[5] [Abboud, Censor-Hillel, Khoury '16]

[6] [Frischknecht, Holzer, Wattenhofer '12]

The CONGEST$_B$ model

How the complexity of existing algorithms scale when more bandwidth is allowed?

Results

There exists an algorithm that solves the APSP problem in $\widetilde{O}(n/B + D)$ rounds.

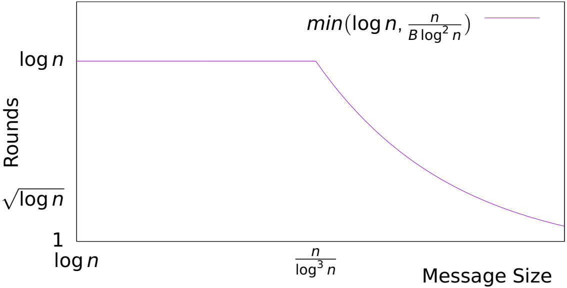
There exists an algorithm that constructs a MST in $\widetilde{O}(D + \sqrt{\frac{n}{B}})$ rounds

There esists an algorithm that finds a $(1+\epsilon)$-approximation of the SSSP problem in $\widetilde{O}(\epsilon^{-O(1)}(\sqrt{\frac{n}{B}} + D))$ rounds.

Results

There is a problem such that:

- It can be solved in $O(\log n)$ rounds with $B = O(\log n)$.
- In order to solve it in less than $\frac{\log n}{2}$ rounds, messages must be of size at least $B = \widetilde{\Omega}(n)$.



Remarks

- In practice, one may prefer to use more bandwidth in order to decrease the latency.
- Different problems scale differently with the bandwidth.
- In some cases more bandwidth does not help.
- It makes sense to analyze algorithms for the whole spectrum of bandwidths.

Open problems

- Analyze more algorithms
- Find common patterns

Thank you