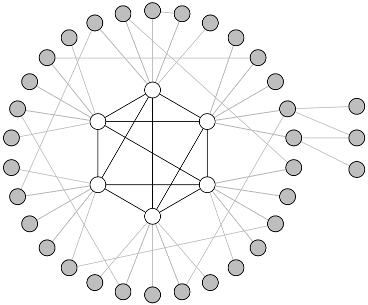
Sparsifying Congested Cliques and Core-Periphery Networks

A. Balliu, P. Fraigniaud, Z. Lotker, and **D. Olivetti**

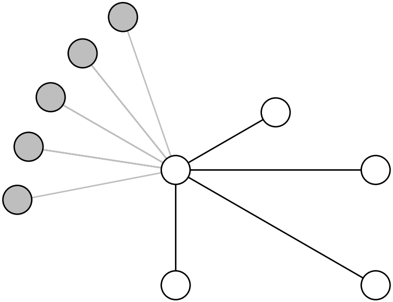
Core-periphery networks

- A novel network architecture for parallel and distributed computing, inspired by social networks and complex systems, proposed by Avin, Borokhovicha, Lotker, and Peleg.
- 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



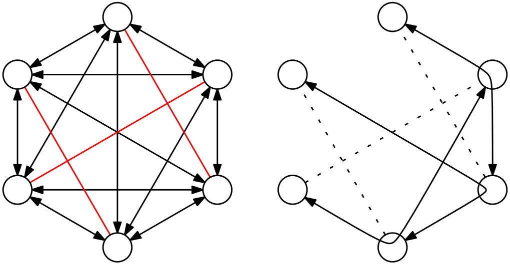
Axiom 1: Core boundary

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.



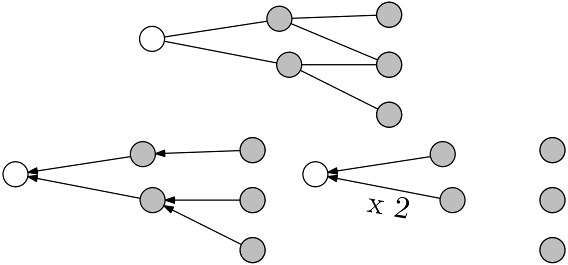
Axiom 2: Clique emulation

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 $v \in C$.



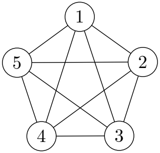
Axiom 3: Periphery-core convergecast

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.



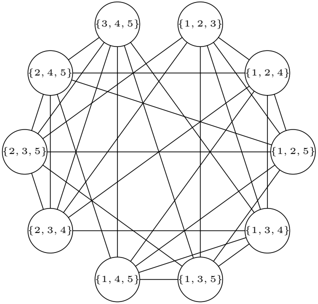
Using 2 rounds to emulate the clique

We want to remove many edges from K_5



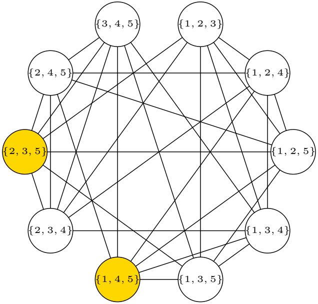
Using 2 rounds to emulate the clique

Consider the Johnson graph $J(n, 3)$



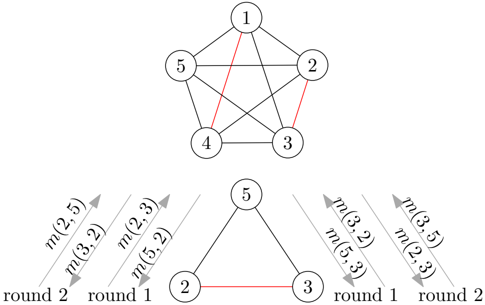
Using 2 rounds to emulate the clique

$$I_0 = \{\{x,y,z\} \in V(J(n,3)) \mid x+y+z \equiv 0 \pmod{n}\}$$



Using 2 rounds to emulate the clique

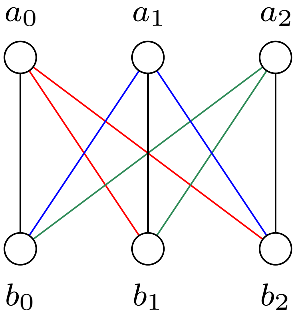
$I_0 = \{\{1,4,5\},\{2,3,5\}\}$. Remove $\{1,4\}$ and $\{2,3\}$.



It is possible to remove approx. $\frac{1}{3}$ of the edges.

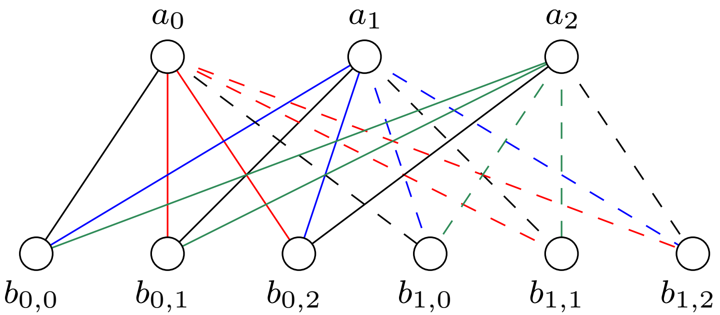
Using more rounds to emulate the clique

The message of b_i is routed to $b_{i'}$ via node a_k where $i + i' + k \equiv 0 \pmod{a}$



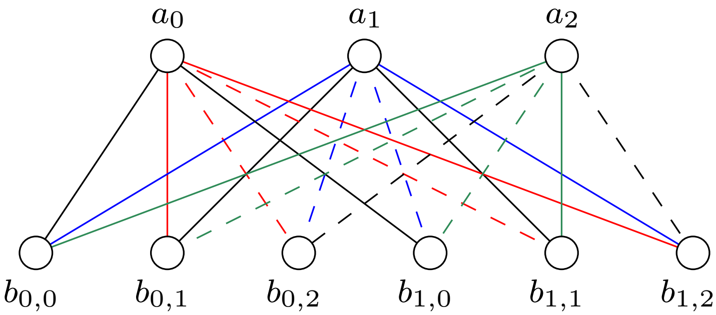
Using more rounds to emulate the clique

Many groups of *b* nodes can do the same concurrently. . .



Using more rounds to emulate the clique

. . . and use the same schema to communicate with other groups.



Using more rounds to emulate the clique

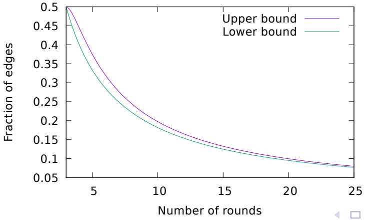
The message of $b_{i,j}$ is routed to $b_{i',j'}$ via node a_k where $j + j' + k \equiv 0 \pmod{a}$ in round $i' - i$.

- This schema requires 2 rounds for each group.
- The communication can be pipelined.
- In total, $\frac{b}{a} + 1$ rounds are required.

Tradeoff between edges and rounds

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.

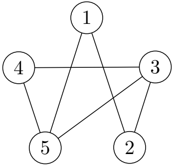
Let $n \geq 1$, $k \in \{1, \ldots, n-1\}$, and let G be an n-node graph that can emulate the n-node clique in k rounds. Then G has at least $\frac{n(n-1)}{k+1}$ edges.

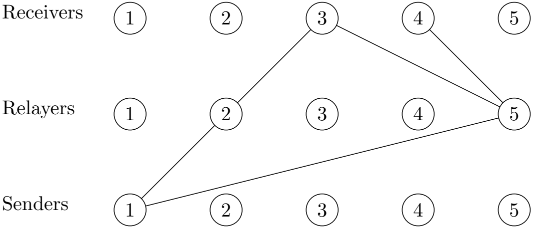


Random graphs $\mathcal{G}_{n,p}$

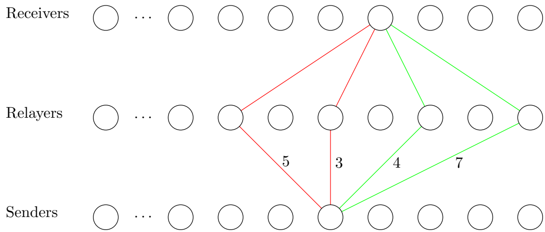
Idea 1: for any missing edge, send the message to a random relayer.

- Assuming independence, it is like balls and bins, every edge has a load of $\frac{1}{p}$ in expectation.
- The most load edge has load $(\frac{1}{p} \times \frac{\log n}{\log \log n})$, bad when p is constant.
- The process is not fully independent

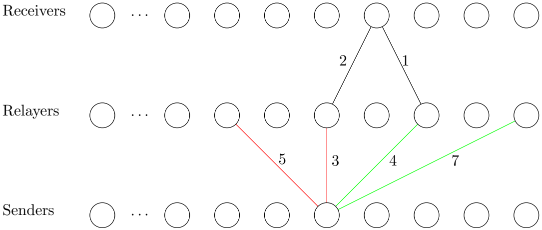




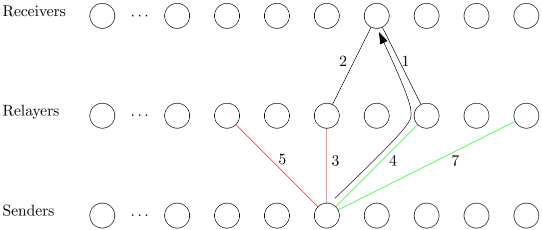
Idea 2: Use the power of many choices



Idea 2: Use the power of many choices



Idea 2: Use the power of many choices



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 $G \in \mathcal{G}_{n,p}$, $\Pr[G \text{ can emulate } K_n \text{ in } O(\min\{\frac{1}{p^2}, np\}) \text{ rounds}] \geq 1 - O(\frac{1}{n^{1+c}})$

Minimum Spanning Tree

MST in the Congest model:

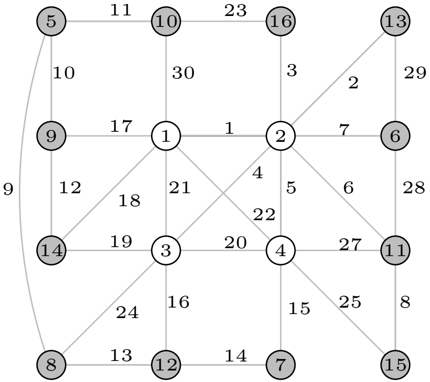
- $D = 1$: $O(\log^* n)$ randomized, $O(\log \log n)$ deterministic
- $D = 2$: $O(\log n)$ deterministic
- $D \geq 3$: $\Omega(\sqrt[3]{n})$
- Core-Periphery ($D \approx 4$): $O(\log^2 n)$ randomized

Minimum Spanning Tree

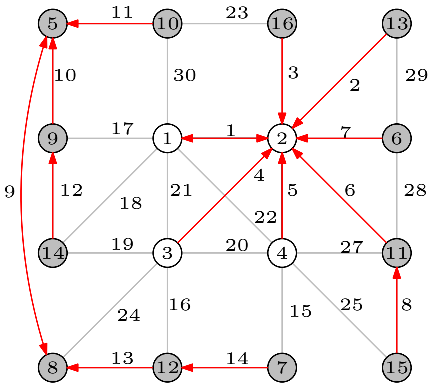
MST in the Congest model:

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

MST by example

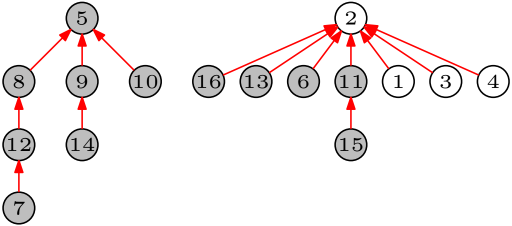


MST by example



MST by example

1	5	8	9
	10	5	11
	9	5	10
	1	2	1
2	16	2	3
	13	2	2
	6	2	7
	2	1	1
3	14	9	12
	8	5	9
	12	8	13
	3	2	4
4	11	2	6
	15	11	8
	7	12	14
	4	2	5



Nodes in the core need to:

1. Find the best edge of each fragment
2. Do pointer jumping and find the root of the merge tree

by avoiding congestion: they can send messages of size $O(\log n)$

Algorithms from the Congested Clique

Lenzen routing protocol

Given a clique of n nodes, if each node is the sender and receiver of $O(n)$ messages, it is possible to exchange the messages in $O(1)$.

Lenzen sorting protocol

Given a clique of n nodes, if each node has $O(n)$ keys, all the $O(n^2)$ keys can be sorted in $O(1)$.

Avoiding congestion

- Sort the edges by tails and find the best edge of each fragment

1	5	1	23
	5	1	17
	1	5	17
2	1	5	23
3	5	1	24
	5	1	16
	1	5	16
4	1	5	15
	5	1	15

$\longrightarrow$

1	5	1	17
	1	5	17
2	1	5	23
3	5	1	16
	1	5	16
4	1	5	15
	5	1	15

$\longrightarrow$

1	1	5	17
	1	5	23
	1	5	16
	1	5	15
2	5	1	17
	5	1	16
	5	1	15

$\longrightarrow$

1	1	5	15
2	5	1	15

Avoiding congestion

- Sort the edges by tails and find the best edge of each fragment
- Sort the remaining edges by their heads to group edges of the merge-tree by common parents

1	1	2
	2	1
	3	2
	4	2
2	5	8
	6	2
	7	12
	8	5
3	9	5
	10	5
	11	2
	12	8
4	13	2
	14	9
	15	11
	16	2

→

1	2	1
	1	2
	3	2
	4	2
2	6	2
	11	2
	13	2
	16	2
3	8	5
	9	5
	10	5
	5	8
4	12	8
	14	9
	15	11
	7	12

} Only 1 request is needed (fragment 2)

Avoiding congestion

- Sort the edges by tails and find the best edge of each fragment
- Sort the remaining edges by their heads to group edges of the merge-tree by common parents
- At this point each node the core (that is of size $O(\sqrt{n})$) has to send and receive $O(\sqrt{n})$, we can use Lenzen routing protocol to perform 1 step of pointer jumping.

Avoiding congestion

- Sort the edges by tails and find the best edge of each fragment
- Sort the remaining edges by their heads to group edges of the merge-tree by common parents
- At this point each node the core (that is of size $O(\sqrt{n})$) has to send and receive $O(\sqrt{n})$, we can use Lenzen routing protocol to perform 1 step of pointer jumping.
- $\log n$ steps of Pointer jumping could be necessary, but they can be deferred to the next phases

Conclusions

- Optimal tradeoff between edges and rounds to emulate the clique.

Conclusions

- Optimal tradeoff between edges and rounds to emulate the clique.
- Clique emulation by random graphs in $O(\frac{1}{p^2})$, can we do better?

Conclusions

- Optimal tradeoff between edges and rounds to emulate the clique.
- Clique emulation by random graphs in $O(\frac{1}{p^2})$, can we do better?
- $O(\log n)$ deterministic algorithm for MST construction, can we do better?

Thank you

References

- C. Avin, M. Borokhovich, Z. Lotker, and D. Peleg. Distributed computing on core-periphery networks: Axiom-based design. In *ICALP (2)*, volume 8573 of *Lecture Notes in Computer Science*, pages 399–410. Springer, 2014.
- Christoph Lenzen. Optimal deterministic routing and sorting on the congested clique. In *PODC*, pages 42–50. ACM, 2013.
- M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104, 2001.