Causality, Influence, and Computation in Possibly Disconnected Synchronous Dynamic Networks

Othon Michail

joint work with
Ioannis Chatzigiannakis
Paul G. Spirakis

Computer Technology Institute & Press "Diophantus" (CTI)
Dept. of Computer Engineering & Informatics (CEID), Univ. of Patras

16th International Conference On Principles Of DIstributed Systems (OPODIS)
December 17-20, 2012
Rome, Italy

Distributed Computation in Worst-case Dynamic Networks

- Distributed computation as usual
  - n processors
  - Interchanging messages with neighbors
- Main Difference:

> The network may change arbitrarily from round to round

- Nodes do not control the changes in the topology
- Of course, not too arbitrarily to prevent any computation
- Should be at least temporally connected

A Model of Network Dynamicity: Dynamic Graphs

- Dynamic graph
  - A sequence $G(1), G(2), \ldots$ of static graphs
  - $G(i)$ is the status of the graph at time/round i
- e.g. a (static) graph is a special case of dynamic graph in which $E(i+1) = E(i)$ for all i

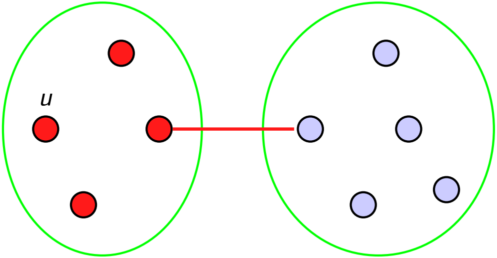
Distributed Computation Model

- Unlimited local storage
- Unique ids of size $O(\log n)$ bits
- Synchronous message passing
  - Discrete steps/rounds
  - Global clock available to the nodes
  - Communication via sending/receiving messages
- Message transmission is broadcast

Connected Instances

[OW05,KLO10]

- $G(i)$ is connected, for all times i
- Implies "good" temporal connectivity
  - The dynamic diameter is $n-1$



Nodes that have heard of u

Nodes that have not heard of u

Possibly Disconnected Instances

Most dynamic networks never have connected instances

- We drop the assumption of connected instances
- We impose weaker conditions to guarantee temporal connectivity

Metrics for Disconnectivity

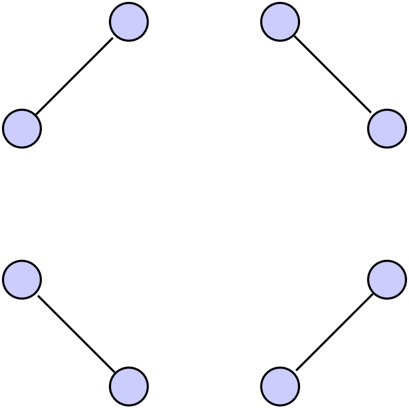
1. Outgoing Influence Time (oit)
2. Incoming Influence Time (iit)
3. Connectivity Time (ct)

Outgoing Influence Time (oit)

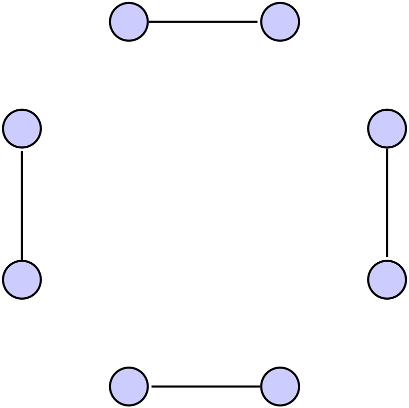
Maximal time until the state of a node influences the state of another node.

- Minimum $k \in \mathbb{N}$ s.t. for all $u \in V$ and all times $t, t' \geq 0$ s.t. $t' \geq t$ it holds that
$$|\mathrm{future}_{(u,t)}(t'+k)| \geq \min\{|\mathrm{future}_{(u,t)}(t')| + 1, n\}$$
- Example: the oit of a dynamic graph with connected instances is 1
- Incoming Influence Time (iit): the same for incoming influences

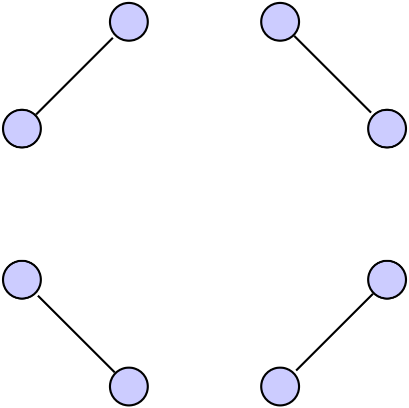
Alternating Matchings



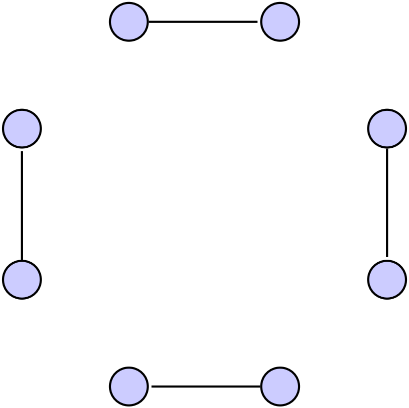
Alternating Matchings



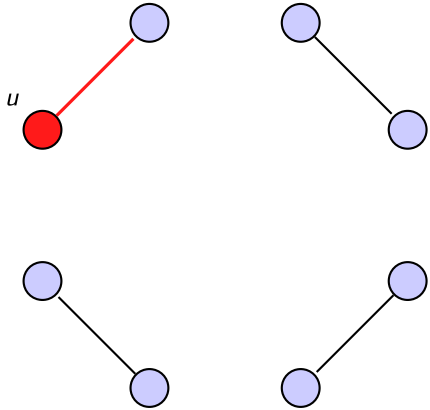
Alternating Matchings



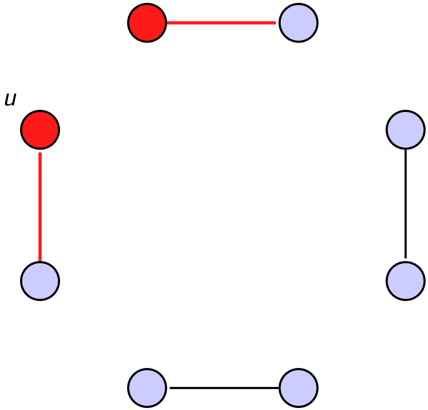
Alternating Matchings



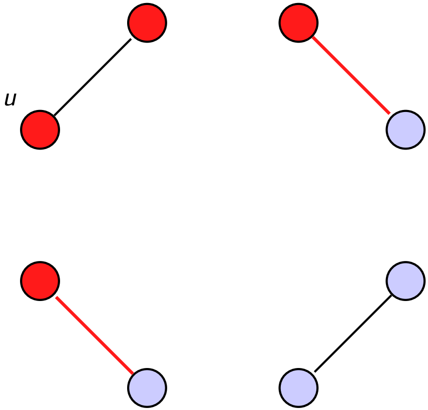
Alternating Matchings (oit=1)



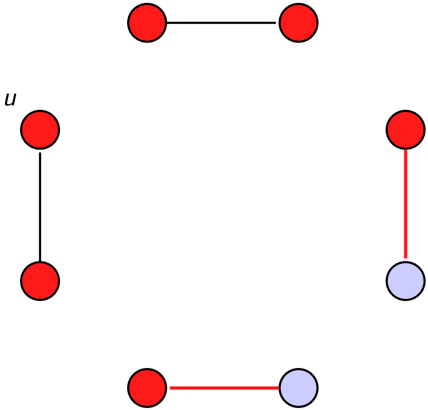
Alternating Matchings (oit=1)



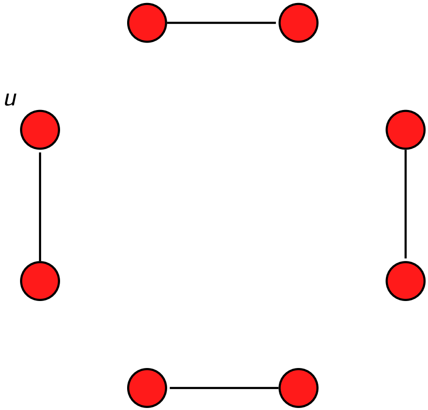
Alternating Matchings (oit=1)



Alternating Matchings (oit=1)



Alternating Matchings (oit=1)

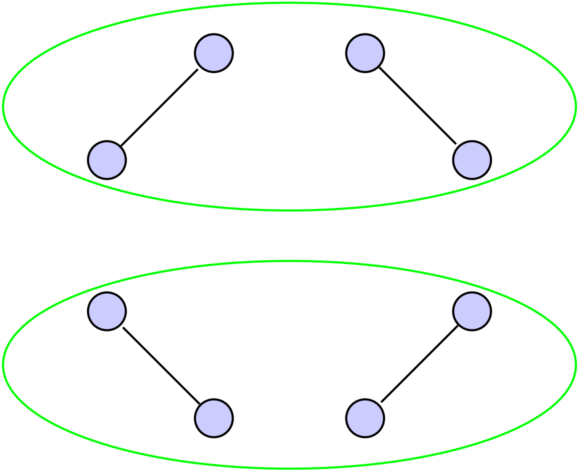


Connectivity Time (ct)

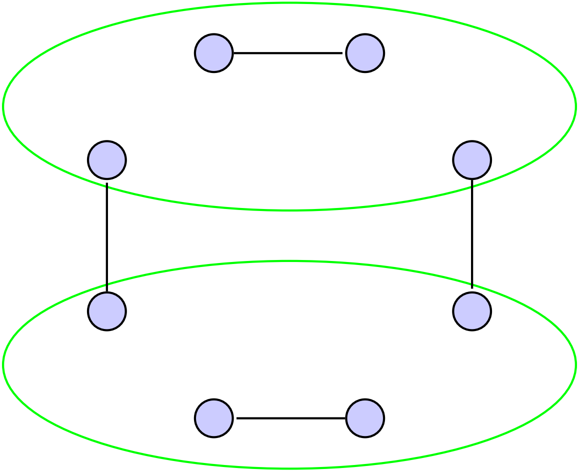
Maximal time until the two parts of any cut of the network become connected.

- Minimum $k \in \mathbb{N}$ s.t. for all times $t \in \mathbb{N}$ the static graph $(V, \bigcup_{i=t}^{t+k-1} E(i))$ is connected
- If the ct is 1 then we obtain a dynamic graph with connected instances
- Greater ct allows for disconnected instances

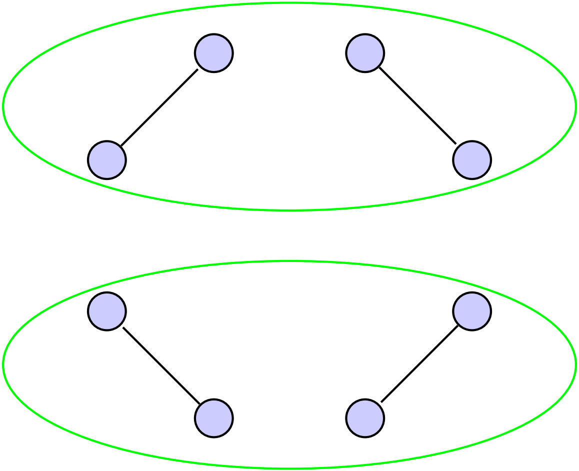
Alternating Matchings (ct=2)



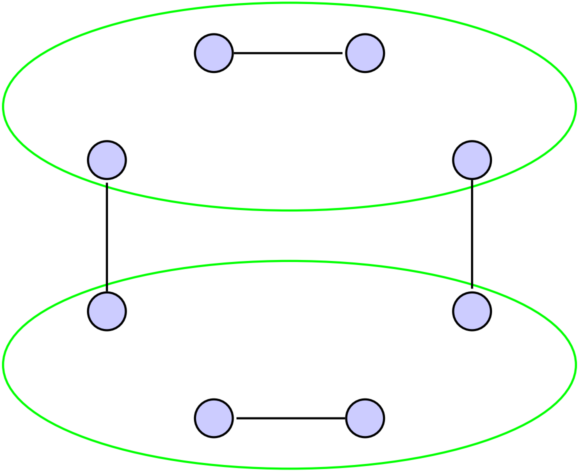
Alternating Matchings (ct=2)



Alternating Matchings (ct=2)



Alternating Matchings (ct=2)



oit vs ct

Proposition

1. *oit* $\leq$ *ct but*
2. *there is a dynamic graph with* *oit* $= 1$ *and ct* $= \Theta(n)$.

Termination Criteria

- To perform global (terminating) computation

> Each node u must be able to tell $\forall 0 \leq t \leq t'$ whether
> $$\mathrm{past}_{(u,t')}(t) = V.$$

- If nodes know n, then a node can determine at time t' whether $\mathrm{past}_{(u,t')}(t) = V$ by counting all different t-states that it has heard of so far
- If n is not known: the subject of our work
- Termination criterion: any locally verifiable property that can be used to determine whether $\mathrm{past}_{(u,t')}(t) = V$

CTI "DIOPHANTUS"

Problems

- Counting: Nodes must determine the network size n
- All-to-all Token Dissemination (or Gossip): each node is provided with a unique token, and all nodes must collect all n tokens
- Functions on Inputs: each node gets an input symbol from some set X and the goal is to have all nodes compute some function f on the distributed input (e.g. min,max,avg)

Termination criteria can be used to directly solve these problems.

Known Upper Bound on the ct

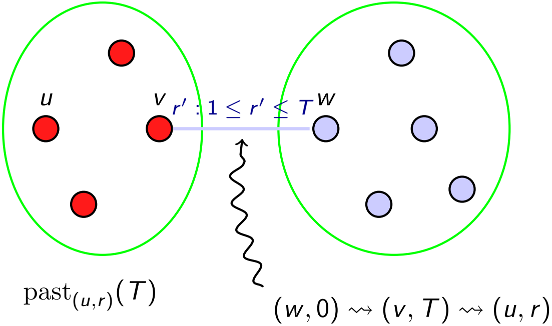
- Nodes know some upper bound T on the ct
- We give an optimal termination criterion
- This gives optimal protocols for our problems

$O(D + T)$ rounds in any dynamic network with dynamic diameter D

Optimal Protocol

Theorem (Repeated Past)

Node u knows at round r that $\mathrm{past}_{(u,r)}(0) = V$ *iff* $\mathrm{past}_{(u,r)}(0) = \mathrm{past}_{(u,r)}(T)$.



Known Upper Bound on the oit

- Nodes know some upper bound K on the oit
- We give a termination criterion which, though being far from the dynamic diameter, is optimal if a node terminates based on its past set
- We then develop a novel technique that gives an optimal termination criterion based on the future set of a node

Inefficiency of Hearing the Past

Theorem

If u has heard of l nodes then it must hear of another node in $O(Kl^2)$ rounds (if an unknown one exists)

- The bound is locally computable
  - K and l are both known
- Poor time complexity: $O(Kn^2)$
- However, some sense of optimality: a node cannot obtain a better upper bound based solely on K and l

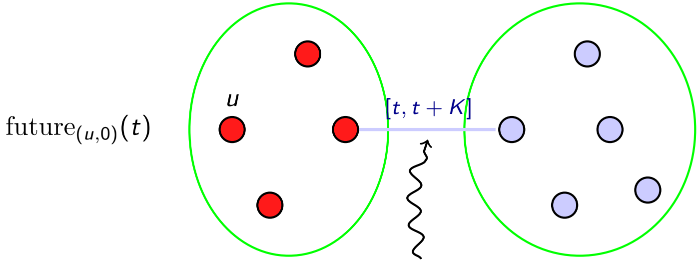
Inefficiency of Hearing the Past

- Even the "Repeated Past" criterion, that is optimal in the ct case, does not work in the oit case
- Essentially, for any t', while u has not been yet causally influenced by all initial states its past set from time 0 may become equal to its past set from time t'

Hearing the Future

- Termination criterion:
  - If $\mathrm{future}_{(u,0)}(t) = \mathrm{future}_{(u,0)}(t+K)$ then $\mathrm{future}_{(u,0)}(t) = V$
- Fundamental goal: Allow a node know its future set
- Novelty: instead of hearing the past, a node now directly keeps track of its future set and is informed by other nodes of its progress

Hearing the Future



- An outgoing influence must occur in at most K rounds
- u keeps track of $\mathrm{future}_{(u,0)}(t)$
- checks whether it has increased by time $t + K$
- If not, no further nodes can exist

Protocol *Hear_from_known*

Theorem

Protocol Hear_from_known terminates in $O(D + K)$ rounds and uses messages of size $O(n \log Kn)$.

- This is optimal w.r.t. time
- Again solves all of our problems

Improving Message Size

- The leader initiates individual conversations with the nodes that it already knows to have been influenced by its initial state
- Sends an invitation to a particular node which is forwarded by all nodes
- A node that receives an invitation replies with the necessary data
  - this message is now preferred and forwarded by all nodes until it gets to the leader
- To make nodes prefer a particular message
  - we accompany messages with timestamps of creation-time and
  - have all nodes prefer the data with the most recent timestamps
- Terminates in $O(Dn^2 + K)$ rounds by using messages of size $O(\log D + \log n)$

Conclusions

- We studied worst-case dynamic networks that are free of any connectivity assumption about their instances
- We proposed new metrics to capture the speed of information propagation
- We proved that fast dissemination and computation are possible even under continuous disconectivity
- We presented optimal termination conditions and protocols based on them for counting and all-to-all dissemination

Research Directions

- Define more informative metrics that capture the speed of propagation
- Develop an asynchronous version of our model in which e.g. nodes broadcast when they detect new neighbors
- Propose methods to reduce the communication complexity
  - So far, nodes broadcast constantly in order to ensure dissemination
- Does visibility or predictability help and to what extend?
- Find better lower and upper bounds for counting and information dissemination
  - Lower bound: $\Omega(nk/\log n)$ (even for centralized algorithms on networks with connected instances and messages of size $O(\log n)$) [DPRSV12]
  - Upper bound: $O(Dn^2 + K)$ (for messages of size $O(\log D + \log n)$)
  - There is a big gap here

Thank You!