# Naming and Counting in Anonymous Unknown 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
Department of Computer Science, University of Liverpool, UK

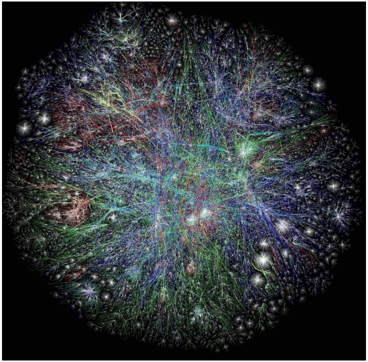
15th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)

November 13-16, 2013

Osaka, Japan

A great variety of systems are dynamic:

- Modern communication networks: inherently dynamic, dynamicity may be of high rate
  - mobile ad hoc, sensor, peer-to-peer, opportunistic, and delay-tolerant networks
- Social networks: social relationships between individuals change, existing individuals leave, new individuals enter



- Transportation networks: transportation units change their positions in the network as time passes
- Physical systems: e.g. systems of interacting particles

# State of the Art

- Traditional communication networks: topology modifications are rare
- The structural and algorithmic properties of dynamic graphs are not well understood yet
- Structural properties of dynamic graphs
  - max-flow min-cut holds with unit capacities [Be, Networks, '96]
  - Menger's theorem violated [KKK, STOC, '00]
  - Analogue of Menger for arbitrary dynamic graphs [MMCS, ICALP, '13]
  - Cost minimization parameters for the design of dynamic networks
- Distributed Computing on Dynamic Networks
  - Worst-case dynamicity [OW, POMC, '05], [KLO, STOC, '10], [MCS, JPDC, '13]
  - Population Protocols (interacting automata) [AADFP, Distr. Comp., '06], [MCS, Book, '11]
  - Randomly Dynamic [CMMPS, PODC, '08], [BCF, Distr. Comp., '11]

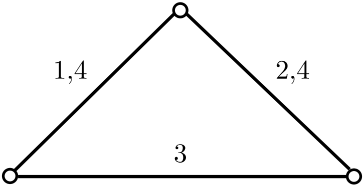
# Model & Problems

- Anonymity: Nodes do not initially have any ids,
- Unknown network: Nodes do not know the topology or the size of the network
- Synchronous message-passing communication
- 2 types of message transmission
  1. Broadcast
  2. One-to-each
- Dynamic graph model: 1-interval connected [KLO, STOC, '10]
- Problems:
  - Counting: Compute $n$
  - Naming: End up with unique identities

## Definition (Dynamic Graph)

A dynamic graph $G$ is a pair $(V, E)$, where $V$ is a set of $n$ nodes and $E : \mathbb{N}_{\geq 1} \rightarrow \mathcal{P}(\{\{u, v\} : u, v \in V\})$ is a function mapping a round number $r$ to a set $E(r)$ of bidirectional links.

- Loosely speaking a graph that changes with time
- Time-labels indicate availability times of edges

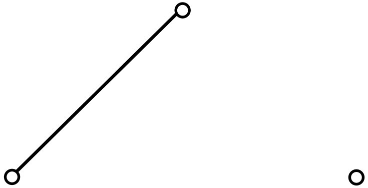


# Dynamic Graph Model

## Definition (Dynamic Graph)

A dynamic graph $G$ is a pair $(V, E)$, where $V$ is a set of $n$ nodes and $E : \mathbb{N}_{\geq 1} \rightarrow \mathcal{P}(\{\{u, v\} : u, v \in V\})$ is a function mapping a round number $r$ to a set $E(r)$ of bidirectional links.

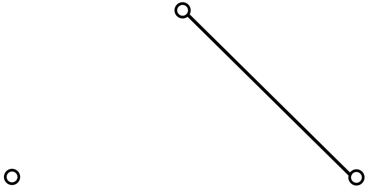
- Loosely speaking a graph that changes with time
- Time-labels indicate availability times of edges



## Definition (Dynamic Graph)

A dynamic graph $G$ is a pair $(V, E)$, where $V$ is a set of $n$ nodes and $E : \mathbb{N}_{\geq 1} \rightarrow \mathcal{P}(\{\{u, v\} : u, v \in V\})$ is a function mapping a round number $r$ to a set $E(r)$ of bidirectional links.

- Loosely speaking a graph that changes with time
- Time-labels indicate availability times of edges

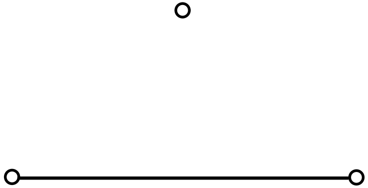


# Dynamic Graph Model

## Definition (Dynamic Graph)

A dynamic graph $G$ is a pair $(V, E)$, where $V$ is a set of $n$ nodes and $E : \mathbb{N}_{\geq 1} \rightarrow \mathcal{P}(\{\{u, v\} : u, v \in V\})$ is a function mapping a round number $r$ to a set $E(r)$ of bidirectional links.

- Loosely speaking a graph that changes with time
- Time-labels indicate availability times of edges

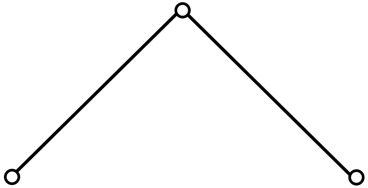


# Dynamic Graph Model

## Definition (Dynamic Graph)

A dynamic graph $G$ is a pair $(V, E)$, where $V$ is a set of $n$ nodes and $E : \mathbb{N}_{\geq 1} \rightarrow \mathcal{P}(\{\{u, v\} : u, v \in V\})$ is a function mapping a round number $r$ to a set $E(r)$ of bidirectional links.

- Loosely speaking a graph that changes with time
- Time-labels indicate availability times of edges



# $T$-interval Connected Dynamic Graphs

- Represent dynamic networks that are connected at every instant
- $T$ represents the rate of connectivity changes

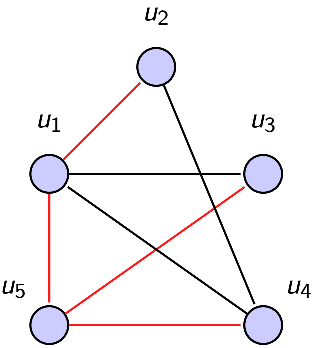
**Definition ([KLO, STOC, '10])**

A dynamic graph $G = (V, E)$ is said to be $T$-interval connected, for $T \geq 1$, if, for all $r \in \mathbb{N}$, the static graph $G_{r,T} := (V, \bigcap_{i=r}^{r+T-1} E(r))$ is connected.

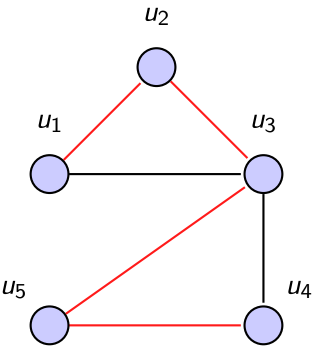
For example

- In 1-interval connected the underlying connected spanning subgraph may change arbitrarily from round to round
- In $\infty$-interval connected a connected spanning subgraph is preserved forever

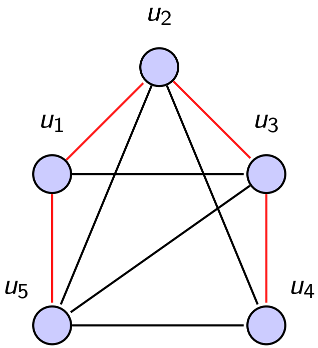
# Example: 1-interval Connected Dynamic Graph



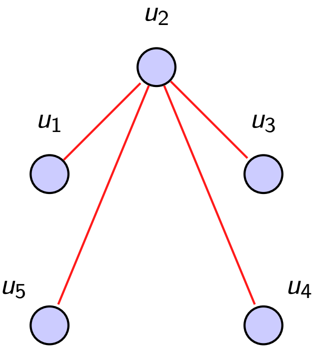
# Example: 1-interval Connected Dynamic Graph



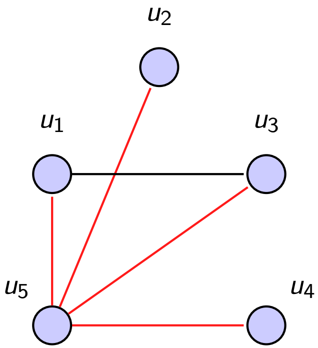
# Example: 1-interval Connected Dynamic Graph



# Example: 1-interval Connected Dynamic Graph

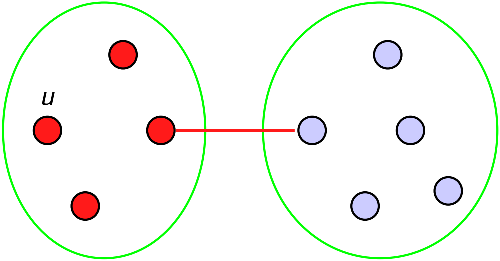


# Example: 1-interval Connected Dynamic Graph



# 1-interval Connected Dynamic Graphs

- Allow for constant propagation of information
- There is always an edge in every cut




Nodes that have heard of $u$

Nodes that have not heard of $u$

# Overview-Contribution

- static networks with broadcast
  - naming: impossible to solve even with a leader and known $n$
  - counting: impossible to solve without a leader
  - counting: with a leader can be solved in linear time
- dynamic networks with broadcast
  - conjecture: impossible to perform nontrivial computation
  - counting upper-bound: can be solved with some additional knowledge (e.g. known upper bound on the maximum degree)
- dynamic networks with one-to-each
  - computationally equivalent to a full-knowledge model

# Static Networks with Broadcast

**Theorem (Naming Impossibility)**

*Naming is impossible to solve by deterministic algorithms in general anonymous (static) networks with broadcast even in the presence of a leader and even if nodes have complete knowledge of the network.*

**Theorem (Counting Impossibility)**

*Without a leader, counting is impossible to solve by deterministic algorithms in general anonymous networks with broadcast.*

- These impossibilities carry over to dynamic networks as well
- Assume a unique leader in order to solve counting

# Counting in Linear Time

- Nodes first compute
  - their distance from the leader and
  - the eccentricity $\epsilon$ of the leader (necessary for termination)
- Each node $u$ knows its number of upper level neighbors $up(u)$

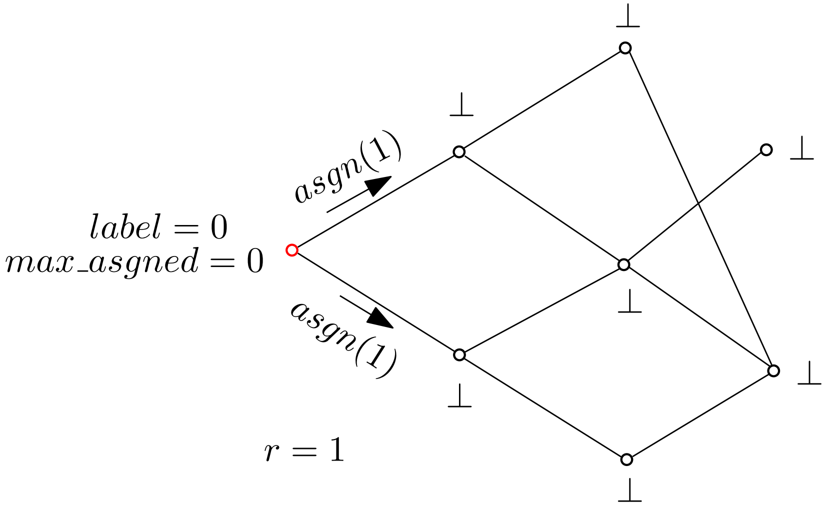
Protocol *Anonymous_Counting*:

- Each lowest-level node $u$ sends to the upper level $1/up(u)$
- Intermediate nodes $v$ sum up the values received from the lower level, add 1, and send the result devided by $up(v)$ (which will be only processed by the upper level)
- The count arrives in parts at the leader, who computes it by summing up
- The leader terminates in $\epsilon + 1$ rounds and the last nodes terminate in $2\epsilon$ rounds

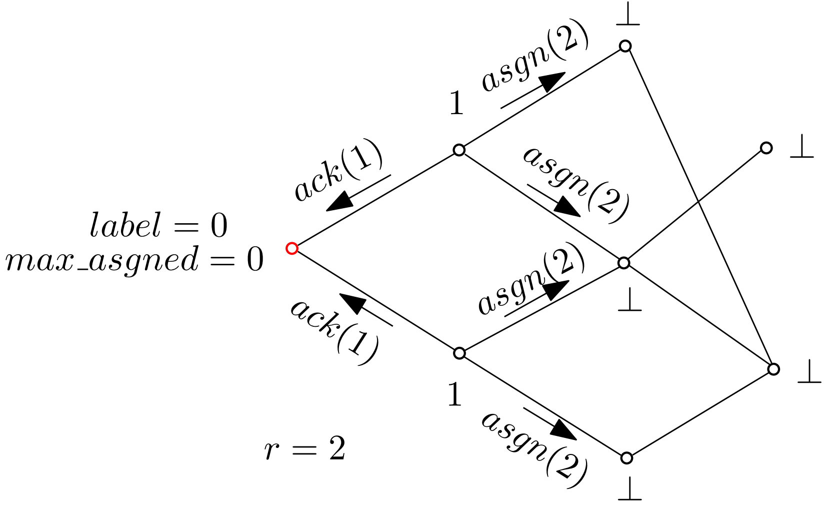
# Counting in Linear Time

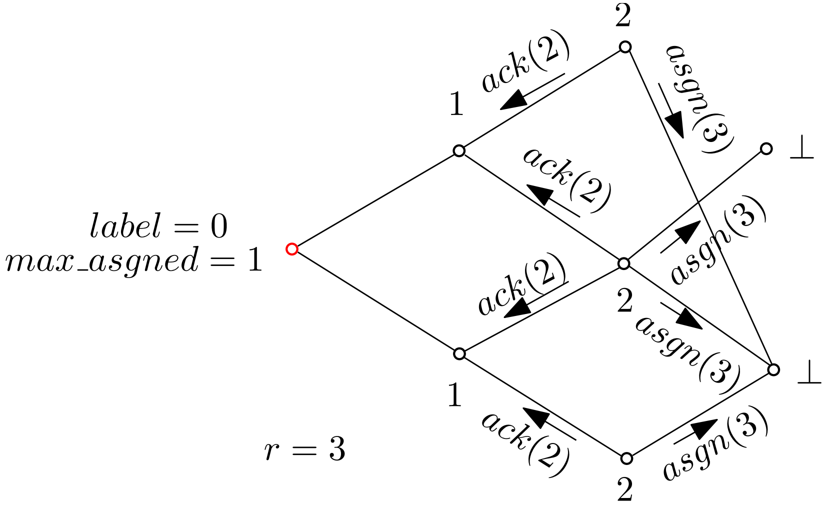
**Theorem**

*Anonymous_Counting solves the counting problem in anonymous static networks with broadcast under the assumption of a unique leader. All nodes terminate in $O(n)$ rounds and use messages of size $O(\log n)$.*

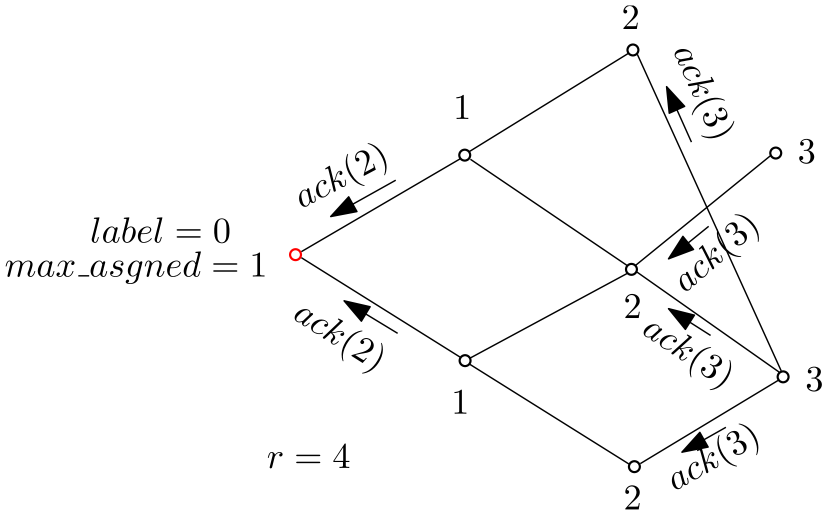


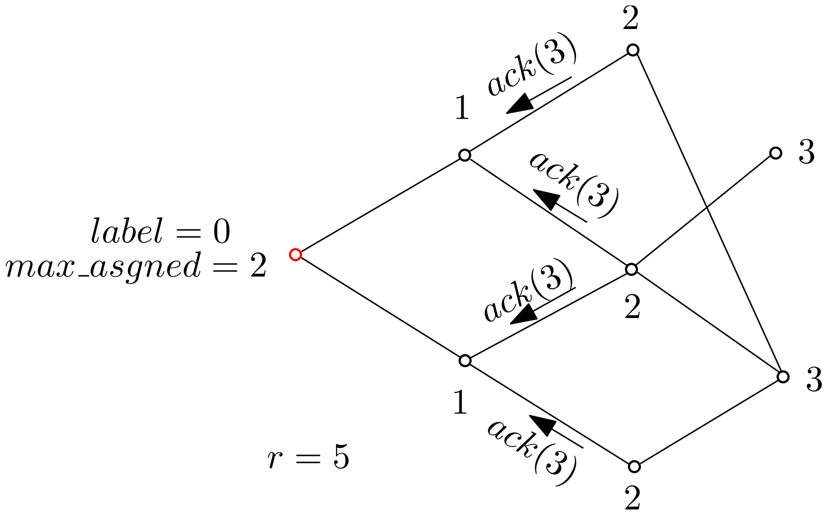
$label = 0$
$max\_asgned = 0$
$asgn(1)$
$asgn(1)$
$\perp$
$r = 1$

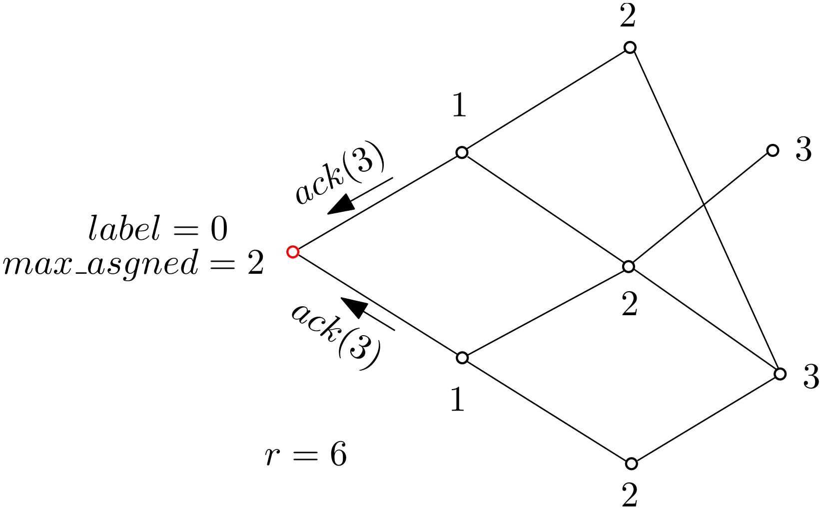

$label = 0$
$max\_asgned = 0$
$ack(1)$
$ack(1)$
1
1
$asgn(2)$
$asgn(2)$
$asgn(2)$
$asgn(2)$
$\perp$
$\perp$
$\perp$
$\perp$
$\perp$
$r = 2$

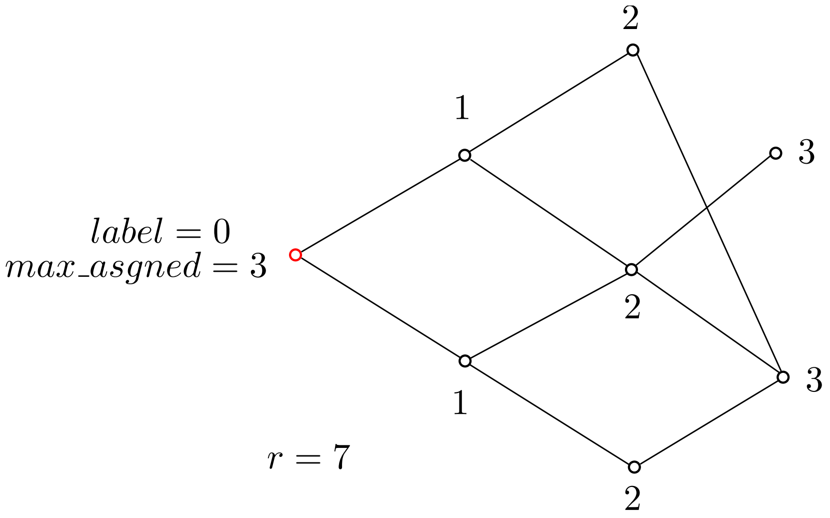

$label = 0$
$max\_asgned = 1$
$r = 3$
1
1
2
2
2
$\perp$
$\perp$
$ack(2)$
$ack(2)$
$ack(2)$
$ack(2)$
$asgn(3)$
$asgn(3)$
$asgn(3)$
$asgn(3)$

# Leader_Eccentricity

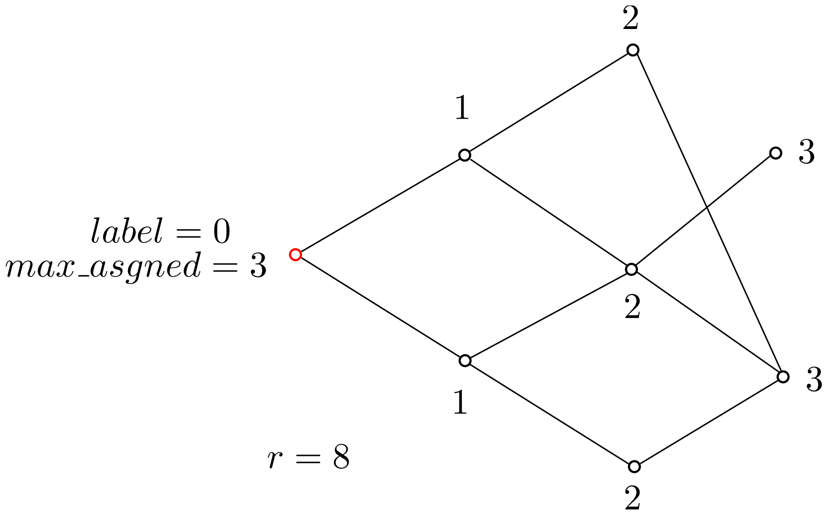



2
ack(3)
1
3
ack(3)
label = 0
max_asgned = 2
ack(3)
2
1
3
ack(3)
r = 5
2


2
1
3
$ack(3)$
$label = 0$
$max\_asgned = 2$
2
$ack(3)$
1
3
$r = 6$
2

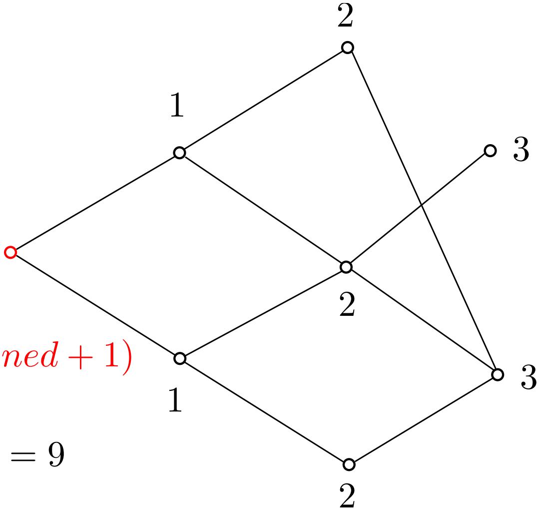

2
1
3
$label = 0$
$max\_asgned = 3$
2
3
1
$r = 7$
2

# Leader_Eccentricity



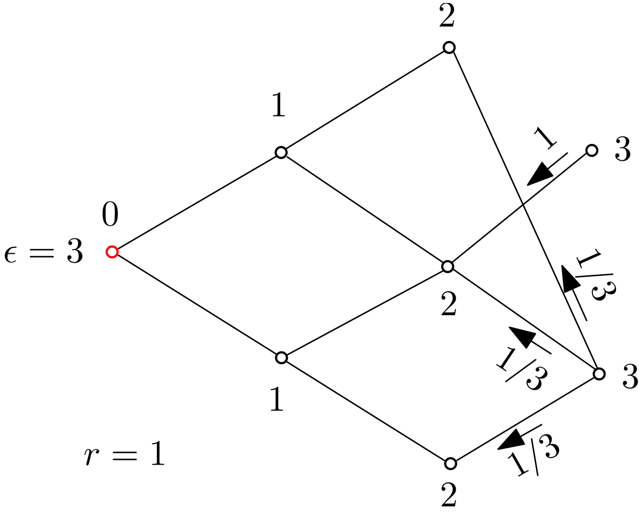
# Leader_Eccentricity

$$label = 0$$
$$max\_asgned = 3$$

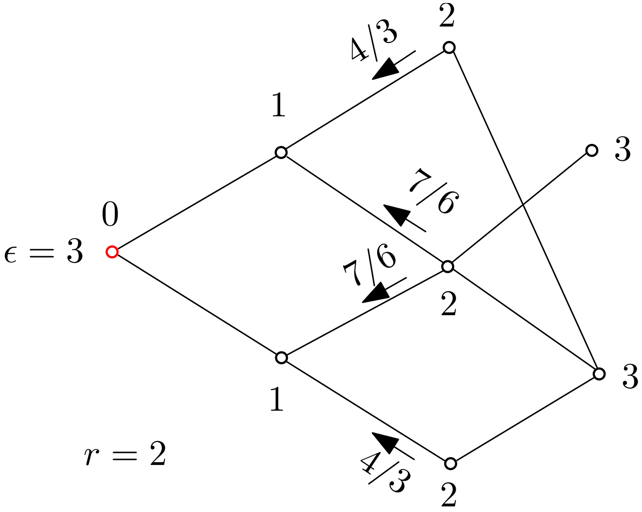


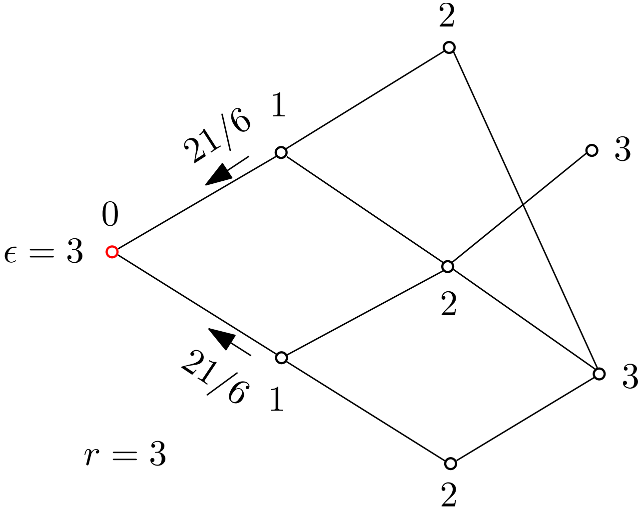

$$r > 2 \cdot (max\_asgned + 1)$$

$$r = 9$$

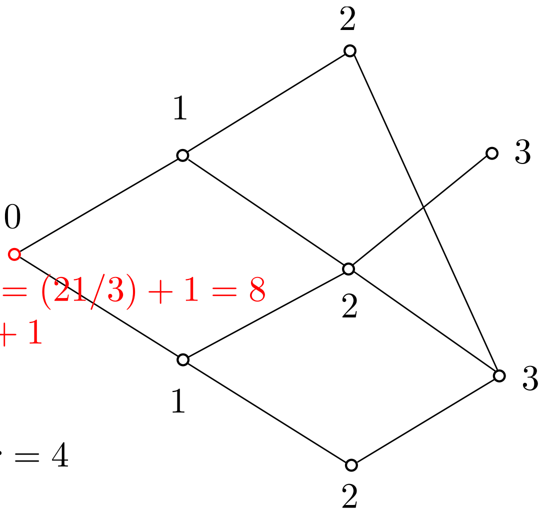


2
1
3
0
$\epsilon = 3$
2
3
1
$r = 1$
2
1
1/3
1/3
1/3


2
4/3
1
3
7/6
0
$\epsilon = 3$
7/6
2
1
3
$r = 2$
4/3
2



2
1
21/6
3
0
$\epsilon = 3$
2
3
21/6
1
$r = 3$
2




$\epsilon = 3$

$count = (21/3) + 1 = 8$

$r = \epsilon + 1$

$r = 4$

# Dynamic Networks with Broadcast

**Conjecture (Impossibility of Nontrivial Computation)**

*It is impossible to compute (even with a leader) the predicate "exists an a in the input".*

- Implies that counting is impossible even with a leader
- Thus, assume a unique leader that knows an upper bound
  1. $d$ on the maximum degree ever to appear in the dynamic network or
  2. $e$ on the maximum expansion (maximum number of concurrent new influences ever occuring)
- We have devised protocols that obtain $O(d^n)$ and $O(n \cdot e)$ upper bounds on the count

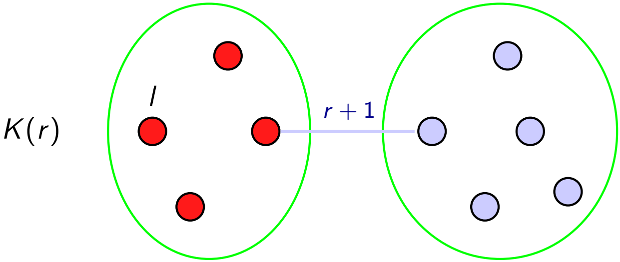
# Dynamic Networks with One-to-Each

- We relax broadcast in order to avoid the previous impossibilities
- One-to-each message transmission
  - In every round $r$, each node $u$ generates a different message $m_{u,v}(r)$ to be delivered to each current neighbor $v$
  - The adversary, in every round $r$, reveals to $u$ a set of locally unique edge-labels $1, 2, \ldots, d_u(r)$
  - Local labels may change arbitrarily from round to round
    - $u$ cannot infer internal states of neighbors based on these labels
- Assume a unique leader
  - Without it impossibility of naming persists even under one-to-each

# Protocol *Dynamic_Naming*

- Already named nodes assign unique ids
- New assignments are acknowledged to the leader (all nodes forward these)
- Nodes that are still unnamed advertize the current round (all nodes forward these)
- At some round $r$, the leader has just updated the set of assigned ids $K(r)$ and a time $t$ at which there existed an unnamed node
- Termination criterion:
  - Iff $|K(r)| \neq |V|$: in at most $|K(r)|$ additional rounds the leader must hear from a node outside $K(r)$, that is either $K(r)$ will change or $t$ will become at least $r$

# Protocol *Dynamic_Naming*



# Protocol *Dynamic_Naming*

- Every node with id *id* assigns names of the form $(id, i)$
  - e.g. $0, (0,1), (0,2), \ldots (0,1,1), (0,1,2), \ldots$
  - Guarantees (inductively) uniqueness of assigned names

Non-leaders

- A node with $id = \perp$, upon receipt of $l$ *assign* messages $(rid_j)$, sets $id \leftarrow \min_j\{rid_j\}$ (in number of bits)
- Upon being assigned id, a node sets $acks \leftarrow acks \cup id$ and sends *ack* $(acks)$ to all its neighbors
- All nodes forward acks
- A node with $id = \perp$ sends *unassigned* $(current\_round)$
- Upon receipt of $l$ *unassigned* messages $(val_j)$ sets $latest\_unassigned \leftarrow \max\{latest\_unassigned, \max_j\{val_j\}\}$ and send *unassigned* $(latest\_unassigned)$

# Protocol *Dynamic_Naming*

The leader

- Upon receipt of $l$ *ack* messages ($acks_j$) if $(\bigcup_j acks_j) \backslash known\_ids \neq \emptyset$ sets
  - $known\_ids \leftarrow known\_ids \cup (\bigcup_j acks_j)$
  - $latest\_new \leftarrow current\_round$
  - $time\_bound \leftarrow current\_round + |known\_ids|$
- upon receipt of $l$ *unassigned* messages ($val_j$) sets
  - $latest\_unassigned \leftarrow \max\{latest\_unassigned, \max_j\{val_j\}\}$
- Termination criterion:
  - If $r > time\_bound$ and $latest\_unassigned < latest\_new$ sends a *halt* $(|known\_ids|)$ message for $|known\_ids| - 1$ rounds and then outputs *id* and halts
  - Any node that receives a *halt* $(n)$ message, sends *halt* $(n)$ for $n - 2$ rounds and then outputs *id* and halts

# Protocol *Dynamic_Naming*

**Theorem**

*Dynamic_Naming solves naming in $O(n)$ rounds using messages of size $\Theta(n^2)$.*

- By executing a simple $O(n)$-time process after *Dynamic_Naming* we can easily reassign minimal (consecutive) names to the nodes
- The leader just floods a list of $(old\_id, new\_id)$ pairs, one for each node in the network

# Protocol *Individual_Conversations*

- Refine *Dynamic_Naming* to reduce the message size to $\Theta(\log n)$

**Theorem**

*Individual_Conversations solves the (minimal) naming problem in $O(n^3)$ rounds using messages of size $\Theta(\log n)$.*

- assigned names are now of the form $k \cdot d + id$
  - $id$ is the id of the node, $d$ is the number of *unique consecutive* ids that the leader knows so far, and $k \geq 1$ is a name counter
- The leader communicates to a remote node $id$ by sending $(id, current\_round)$
- The timestamp allows all nodes to prefer the latest message

# Protocol *Individual_Conversations*

- Gain: the message is delivered and no node ever issues a message containing more than one id
- The remote node then can reply in the same way
- For the assignment formula to work
  - nodes that obtain ids are not allowed to further assign ids until the leader freezes all named nodes and reassigns unique consecutive ids
  - During freezing, the leader is informed of any new assignments by the named nodes and terminates if all report that no further assignments were performed

# Techniques

The techniques developed here are valuable in their own right:

- Hearing the Future (instead of the past)
  - has given the first time-optimal protocols for counting and token-dissemination in dynamic networks that are possibly disconnected at every instant [MCS, JPDC, '13]
- Individual Conversations
  - has given the first bit-optimal protocols [MCS, JPDC, '13]

# Open Problems

- Prove our conjecture
- Find a faster protocol for naming in dynamic networks with one-to-each using messages of size $\Theta(\log n)$
  - *Individual_Conversations* needs $O(n^3)$ rounds
- Find lower bounds
- Consider the same problems in possibly disconnected dynamic networks [MCS, JPDC, '13]

# Open Problems

- Information dissemination is only guaranteed under continuous broadcasting
  - How can the number of redundant transmissions be reduced in order to improve communication efficiency?
  - Is there a way to exploit visibility to this end?
  - Does predictability help (i.e. some knowledge of the future)?

# Thank You!