Interconnection network with a shared whiteboard

Ivan Rapaport

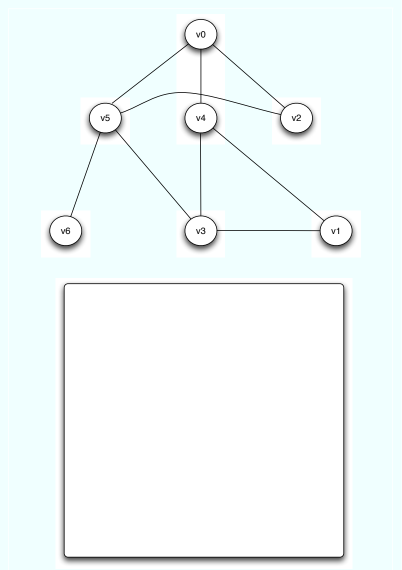
(joint work with F. Becker, A. Kosowski, N. Nisse and K. Suchan)

Universidad de Chile

Outline

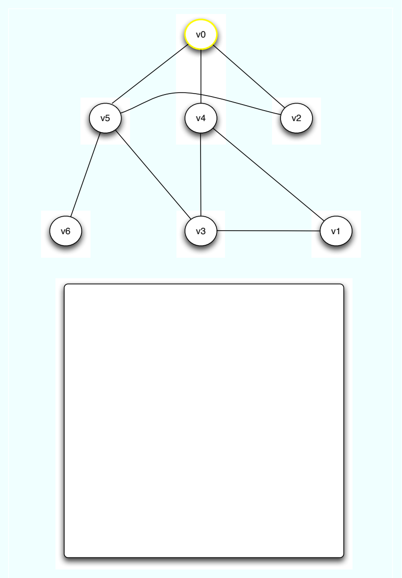
BFS-TREE

FREESYNC



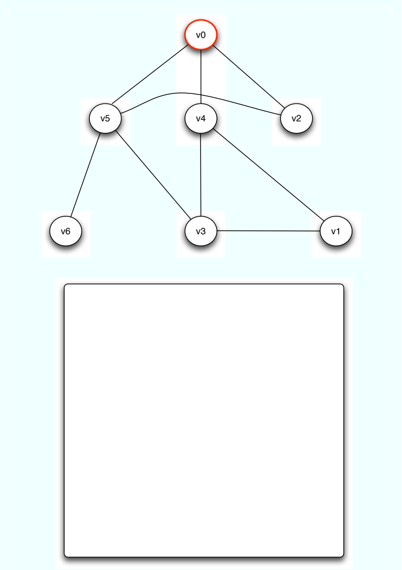
BFS-TREE

FREESYNC



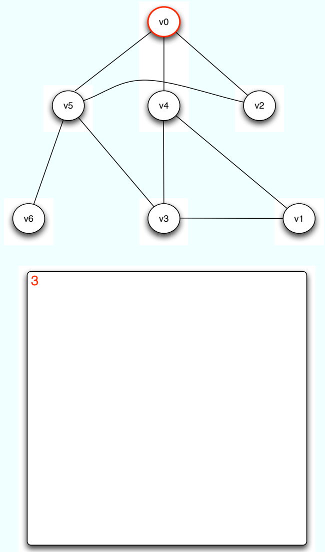
BFS-tree

FreeSync



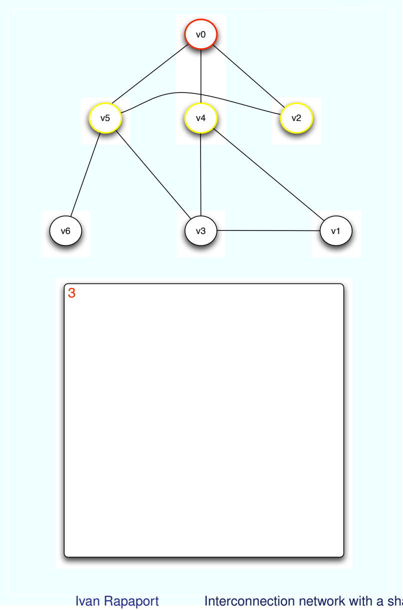
BFS-tree

FreeSync

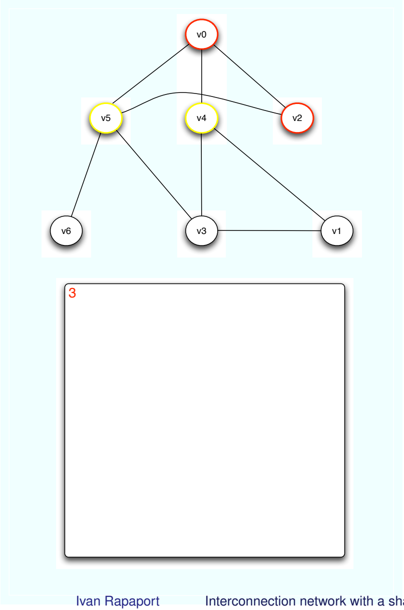


BFS-TREE

FREESYNC

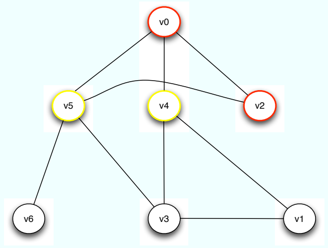


BFS-TREE
FREESYNC



BFS-TREE

FREESYNC

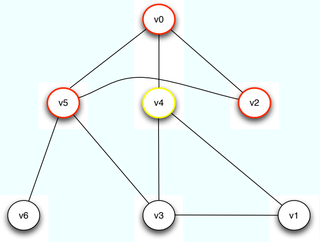


3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

BFS-tree

FreeSync

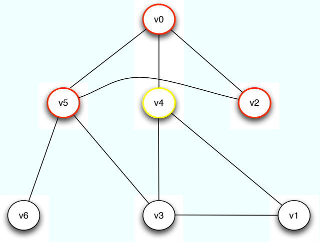


3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

BFS-TREE

FREESYNC



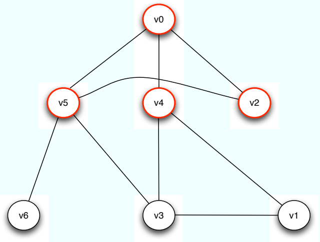
3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

BFS-tree

FreeSync



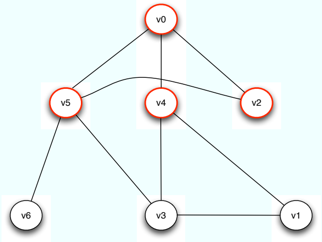
3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

BFS-TREE

FREESYNC



3

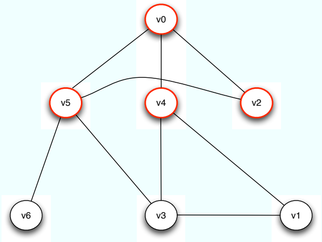
id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

BFS-tree

FreeSync



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

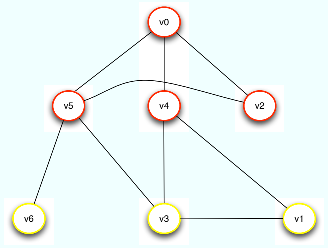
id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

BFS-tree

FreeSync



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

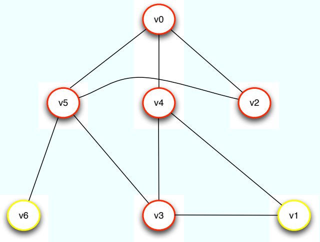
id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

BFS-TREE

FREESYNC



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

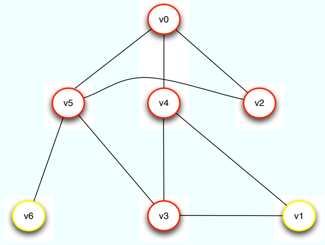
id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

BFS-TREE

FREESYNC



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

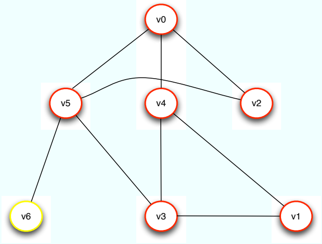
id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

id 3 , lay 2, fa 4, #lay(1) 2, #lay(-1) 1, 0

BFS-tree

FreeSync



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

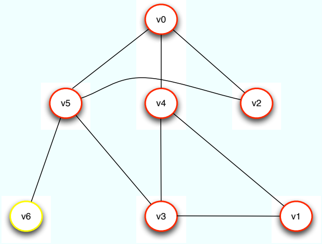
id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

id 3 , lay 2, fa 4, #lay(1) 2, #lay(-1) 1, 0

BFS-TREE

FREESYNC



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

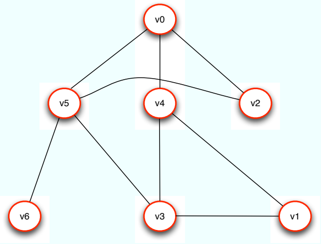
(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

id 3 , lay 2, fa 4, #lay(1) 2, #lay(-1) 1, 0

id 1 , lay 2, fa 4, #lay(1) 1, #lay(-1) 1, 1

BFS-TREE

FREESYNC



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

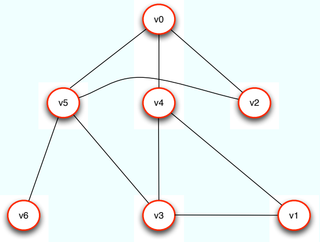
(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

id 3 , lay 2, fa 4, #lay(1) 2, #lay(-1) 1, 0

id 1 , lay 2, fa 4, #lay(1) 1, #lay(-1) 1, 1

BFS-TREE

FREESYNC



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

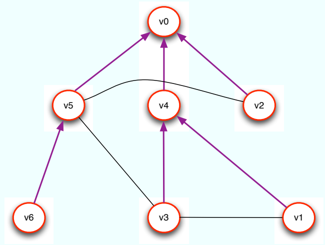
id 3 , lay 2, fa 4, #lay(1) 2, #lay(-1) 1, 0

id 1 , lay 2, fa 4, #lay(1) 1, #lay(-1) 1, 1

id 6 , lay 2, fa 5, #lay(1) 1, #lay(-1) 0, 0

BFS-tree

FreeSync



3

id 2 , lay 1, fa 0, #lay(0) 1, #lay(-0) 1, 0

id 5 , lay 1, fa 0, #lay(0) 1, #lay(-0) 3, 1

id 4 , lay 1, fa 0, #lay(0) 1, #lay(-0) 2, 0

(1+1+1 = 3 and 1+3+2-2(0+1+0) = 4)

id 3 , lay 2, fa 4, #lay(1) 2, #lay(-1) 1, 0

id 1 , lay 2, fa 4, #lay(1) 1, #lay(-1) 1, 1

id 6 , lay 2, fa 5, #lay(1) 1, #lay(-1) 0, 0

(2+1+1 = 4 and 1+1-2(0+1+0) = 0)

Connectivity

FreeAsync

We have shown that in the previous, FreeSync model, we can solve the Connectivity problem.

CONNECTIVITY

FREEASYNC

We have shown that in the previous, FREESYNC model, we can solve the CONNECTIVITY problem.

In fact, if the number of arcs crossing from some layer i to layer $i + 1$ is 0 and **if there are still some nodes which have not raised their hand yet** then the graph can not be connected (the size n of the network is a known parameter).

CONNECTIVITY
FREEASYNC

We have shown that in the previous, FREESYNC model, we can solve the CONNECTIVITY problem.

In fact, if the number of arcs crossing from some layer i to layer $i + 1$ is 0 and **if there are still some nodes which have not raised their hand yet** then the graph can not be connected (the size n of the network is a known parameter).

Consider now the weaker, FREEASYNC model, where the nodes **create their messages as soon as they raise their hands** (before being chosen by the adversary).

CONNECTIVITY
FREEASYNC

We have shown that in the previous, FREESYNC model, we can solve the CONNECTIVITY problem.

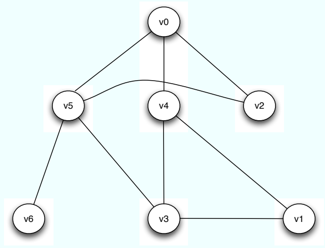
In fact, if the number of arcs crossing from some layer i to layer $i+1$ is 0 and **if there are still some nodes which have not raised their hand yet** then the graph can not be connected (the size n of the network is a known parameter).

Consider now the weaker, FREEASYNC model, where the nodes **create their messages as soon as they raise their hands** (before being chosen by the adversary).

We are going to see that, in this weaker FREEASYNC model, we can construct a SPANNING TREE **when the graph is connected**.

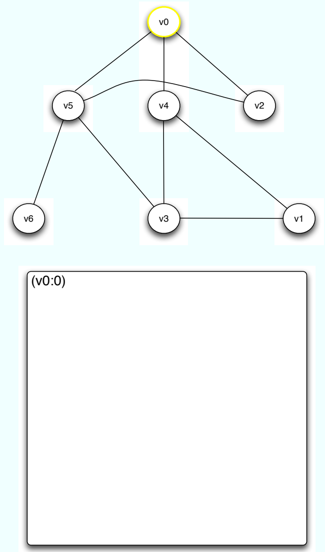
SPANNING TREE

FREEASYNC



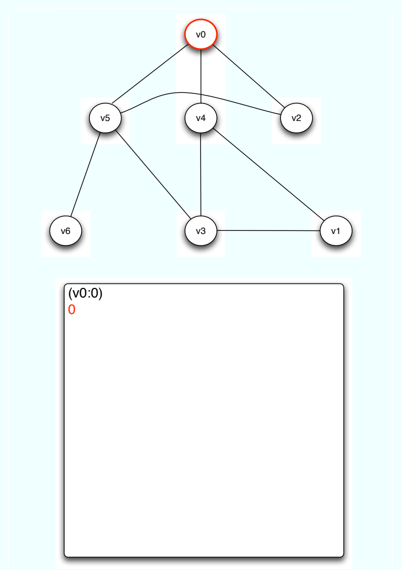
Spanning Tree

FreeAsync



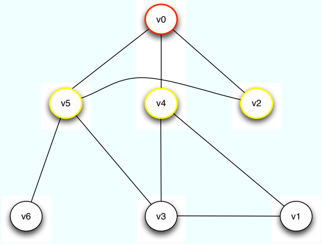
Spanning Tree

FreeAsync



SPANNING TREE

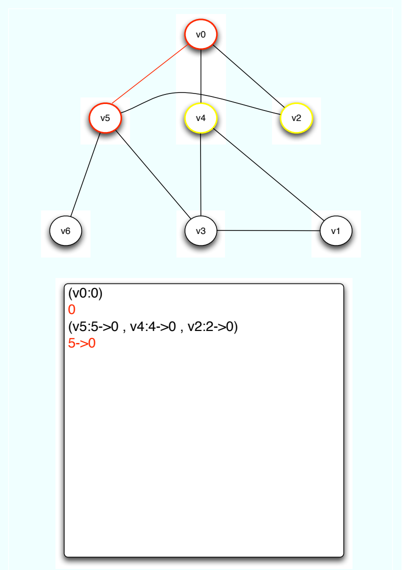
FREEASYNC



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)

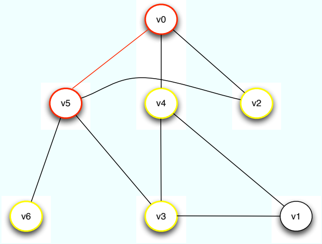
Spanning Tree

FreeAsync



SPANNING TREE

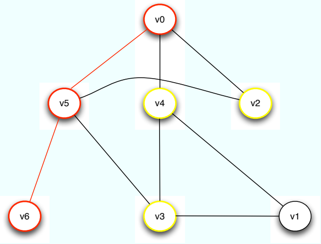
FREEASYNC



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)

SPANNING TREE

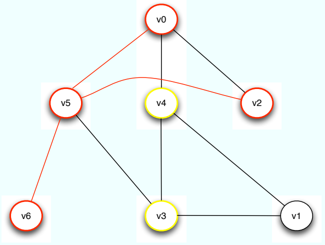
FREEASYNC



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5

Spanning Tree

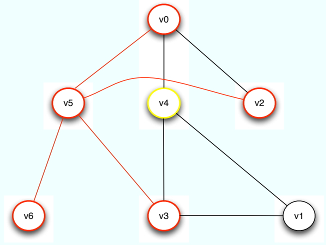
FreeAsync



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5
2->5

Spanning Tree

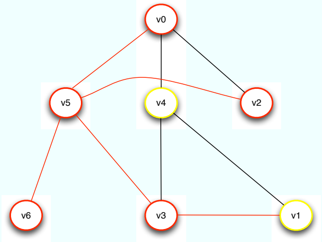
FreeAsync



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5
2->5
3->5

Spanning Tree

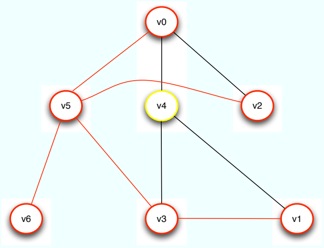
FreeAsync



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5
2->5
3->5
(1->3)

Spanning Tree

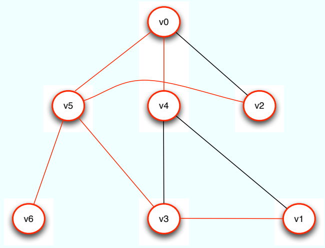
FreeAsync



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5
2->5
3->5
(1->3)
1->3

Spanning Tree

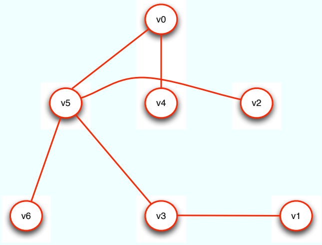
FreeAsync



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5
2->5
3->5
(1->3)
1->3
4->0

Spanning Tree

FreeAsync



(v0:0)
0
(v5:5->0 , v4:4->0 , v2:2->0)
5->0
(v6:6->5 , v3:3->5)
6->5
2->5
3->5
(1->3)
1->3
4->0

CONNECTIVITY

FREEASYNC

Is it possible to decide, in this weaker FREEASYNC model, whether the graph is connected?

CONNECTIVITY

FREEASYNC

Is it possible to decide, in this weaker FREEASYNC model, whether the graph is connected?

Is it true that FREEASYNC < FREESYNC?

CONNECTIVITY
FREEASYNC

Is it possible to decide, in this weaker FREEASYNC model, whether the graph is connected?

Is it true that FREEASYNC < FREESYNC?

Consider now the even weaker, SIMSYNC model, where the nodes **create the messages when they are chosen by the adversary** but **they all raise their hands in the beginning**.

CONNECTIVITY
FREEASYNC

Is it possible to decide, in this weaker FREEASYNC model, whether the graph is connected?

Is it true that FREEASYNC < FREESYNC?

Consider now the even weaker, SIMSYNC model, where the nodes **create the messages when they are chosen by the adversary** but **they all raise their hands in the beginning**.

We already know that SIMASYNC < FREEASYNC.

CONNECTIVITY
FREEASYNC

Is it possible to decide, in this weaker FREEASYNC model, whether the graph is connected?

Is it true that FREEASYNC < FREESYNC?

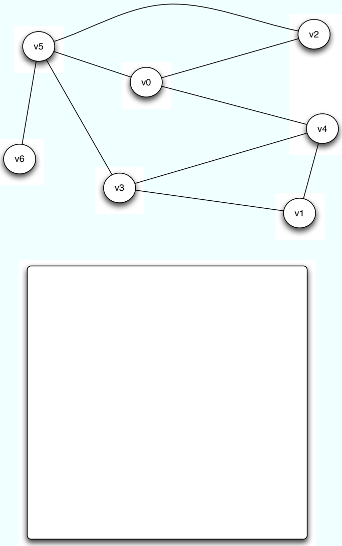
Consider now the even weaker, SIMSYNC model, where the nodes **create the messages when they are chosen by the adversary** but **they all raise their hands in the beginning**.

We already know that SIMASYNC < FREEASYNC.

We are going to show now that we can solve the maximal independent set problem (MIS) in this SIMASYNC model.

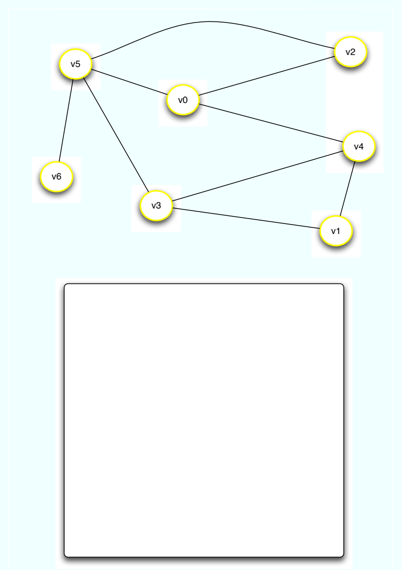
MIS(v_0)

SimSync



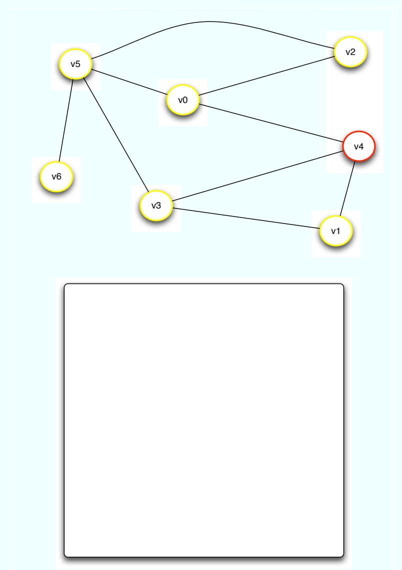
MIS(v_0)

SIMSYNC



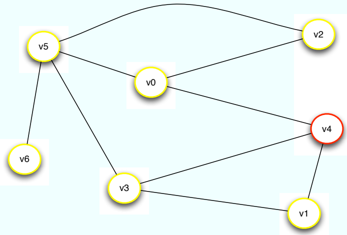
MIS(v_0)

SIMSYNC



MIS(v_0)

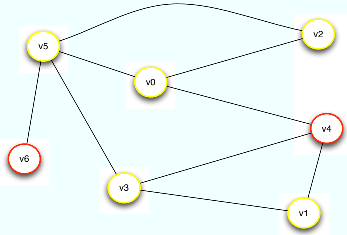
SIMSYNC



4, no

MIS(v_0)

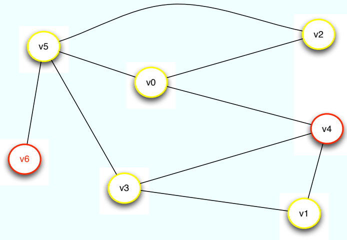
SimSync



4, no

MIS(v_0)

SIMSYNC

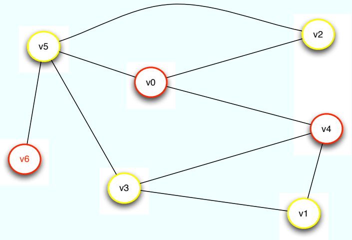


4, no

6, yes

MIS(v_0)

SimSync

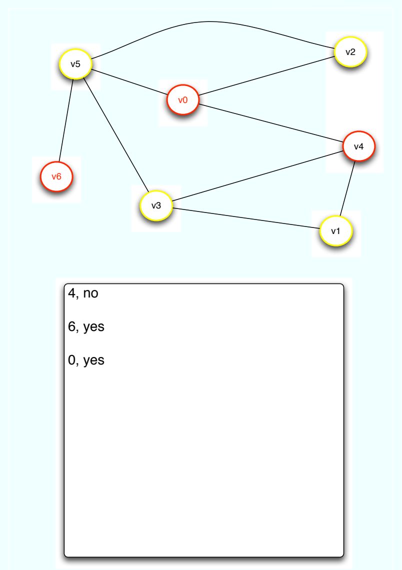


4, no

6, yes

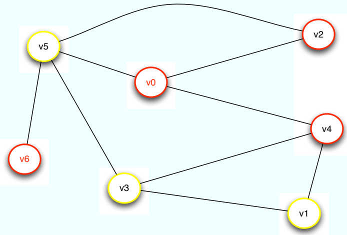
MIS(v_0)

SIMSYNC



MIS(v_0)

SimSync



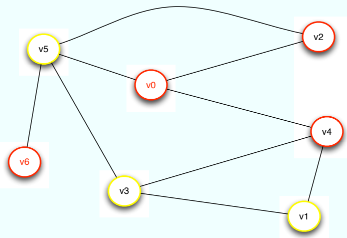
4, no

6, yes

0, yes

MIS(v_0)

SIMSYNC



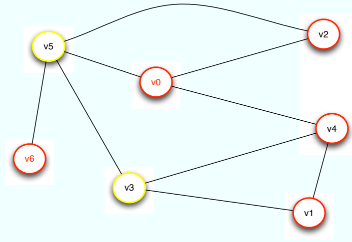
4, no

6, yes

0, yes

2, no

MIS(v_0)
SimSync



4, no

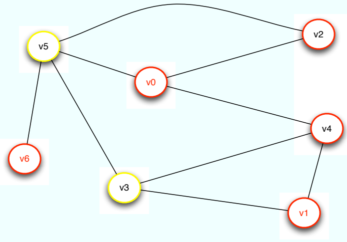
6, yes

0, yes

2, no

MIS(v_0)

SIMSYNC



4, no

6, yes

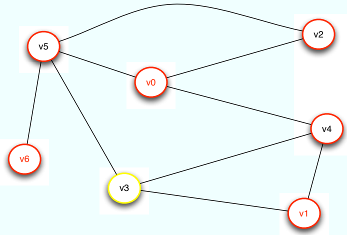
0, yes

2, no

1, yes

MIS(v_0)

SIMSYNC



4, no

6, yes

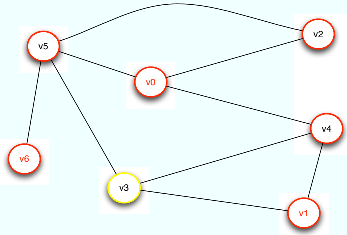
0, yes

2, no

1, yes

MIS(v_0)

SimSync



4, no

6, yes

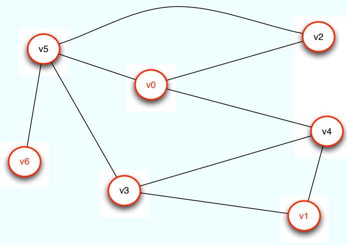
0, yes

2, no

1, yes

5, no

MIS(v_0)
SIMSYNC



4, no

6, yes

0, yes

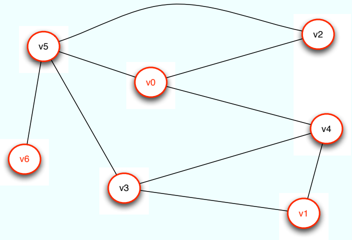
2, no

1, yes

5, no

MIS(v_0)

SimSync



4, no

6, yes

0, yes

2, no

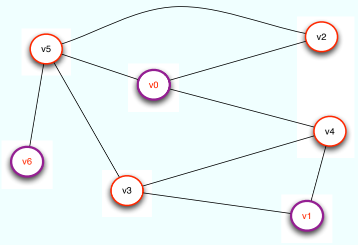
1, yes

5, no

3, no

MIS(v_0)

SimSync



4, no

6, yes

0, yes

2, no

1, yes

5, no

3, no

MIS(v_0)
SimAsync

The weakest model, SimAsync, is the one presented by Ioan Todinca 15 minutes ago, where all nodes raise their hand in the beginning and they create their messages also in the beginning (**no one reads the whiteboard**).

MIS(v_0)

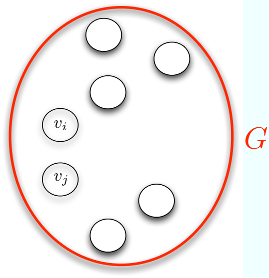
SimAsync

The weakest model, SimAsync, is the one presented by Ioan Todinca 15 minutes ago, where all nodes raise their hand in the beginning and they create their messages also in the beginning (**no one reads the whiteboard**).

By proving that MIS(v_0) can not be solved in this weakest model, we are going to conclude that SimAsync < SimSync.

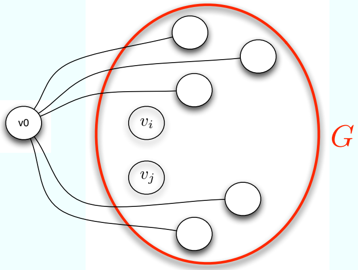
MIS(v_0)

SimAsync



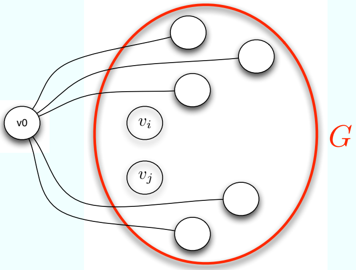
MIS(v_0)

SimAsync



MIS(v_0)

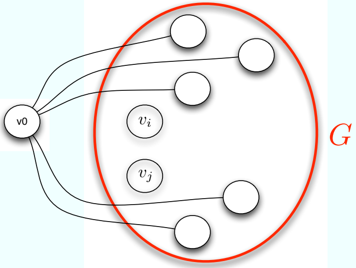
SIMASYNC



$$v_i v_j \in E(G) \iff output = \{v_0, v_i, v_j\}$$

MIS(v_0)

SimAsync

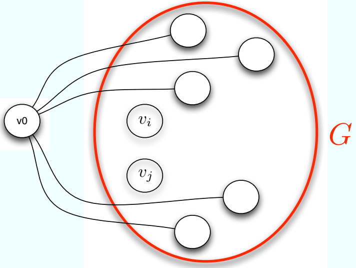


$$v_i v_j \in E(G) \iff output = \{v_0, v_i, v_j\}$$

m_1
$\vdots$
m_{i-1}
M_i
m_{i+1}
$\vdots$
m_{j-1}
M_j
m_{j+1}
$\vdots$
m_n

MIS(v_0)

SimAsync



$$v_i v_j \in E(G) \iff output = \{v_0, v_i, v_j\}$$

m_1	
$\vdots$	
m_{i-1}	
	M_i
m_{i+1}	
$\vdots$	
m_{j-1}	
	M_j
m_{j+1}	
$\vdots$	
m_n	

m_1	M_1
$\vdots$	$\vdots$
m_{i-1}	M_{i-1}
m_i	M_i
m_{i+1}	M_{i+1}
$\vdots$	$\vdots$
m_{j-1}	M_{j-1}
m_j	M_j
m_{j+1}	M_{j+1}
$\vdots$	$\vdots$
m_n	M_n

MIS(v_0)
SIMASYNC

There are $2n$ messages of size $O(\log n)$ each.

MIS(v_0)
SIMASYNC

There are $2n$ messages of size $O(\log n)$ each.

Therefore it could be possible to reconstruct any n-vertex graph with $O(n \log n)$ bits.

MIS(v_0)
SIMASYNC

There are $2n$ messages of size $O(\log n)$ each.

Therefore it could be possible to reconstruct any n-vertex graph with $O(n \log n)$ bits.

But there are $2^{\binom{n}{2}}$ different n-vertex graphs. Therefore we would need $\binom{n}{2}$ bits to encode them. $\Rightarrow\Leftarrow$