Local Checkability, No Strings Attached

Klaus-Tycho Förster, *Thomas Lüdi, Jochen Seidel, Roger Wattenhofer*

January 06, 2016 @ ICDCN 2016 - Singapore

Deciding

Towards Robust Distributed Systems

Eric A. Brewer
UC Berkeley and Inktomi

Current distributed systems, even the ones that work, tend to be very fragile: they are hard to keep up, hard to manage, hard to grow, hard to evolve, and hard to program. In this talk, I look at several issues in an attempt to clean up the way we think about these systems. These issues include the fault model, high availability, graceful degradation, data consistency, evolution, composition, and autonomy.

These are not (yet) provable principles, but merely ways to think about the issues that simplify design in practice. They draw on experience at Berkeley and with giant-scale systems built at Inktomi, including the system that handles 50% of all web searches.

Prove

Deciding vs Checking

Towards Robust Distributed Systems

Eric A. Brewer
UC Berkeley and Inktomi

Current distributed systems, even the ones that work, tend to be very fragile: they are hard to keep up, hard to manage, hard to grow, hard to evolve, and hard to program. In this talk, I look at several issues in an attempt to clean up the way we think about these systems. These issues include the fault model, high availability, graceful degradation, data consistency, evolution, composition, and autonomy.

These are not (yet) provable principles, but merely ways to think about the issues that simplify design in practice. They draw on experience at Berkeley and with giant-scale systems built at Inktomi, including the system that handles 50% of all web searches.

Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services

Seth Gilbert and Nancy Lynch
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
sethg@mit.edu,lynch@theory.lcs.mit.edu

Abstract

When designing distributed web services, there are three properties that are commonly desired: consistency, availability, and partition tolerance. It is impossible to achieve all three. In this note, we prove this conjecture in the asynchronous network model, and then discuss solutions to this dilemma in the partially synchronous model.

1 Introduction

At PODC 2000, Brewer[1], in an invited talk [2], made the following conjecture: it is impossible for a web service to provide the following three guarantees:

- Consistency
- Availability
- Partition-tolerance

Prove **Verify**

Complexity Theory

P

NP

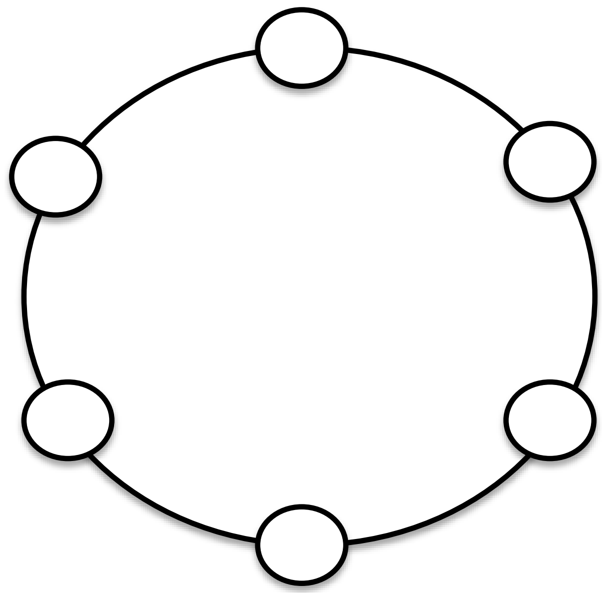
Prove
In polynomial time

Verify
In polynomial time

Overview

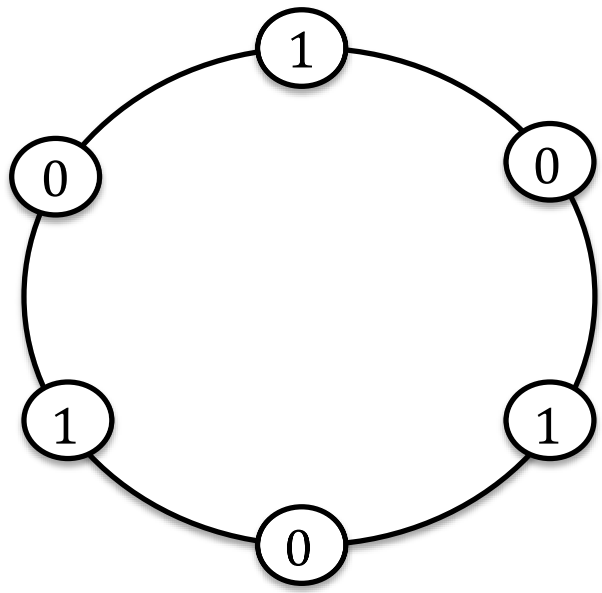
- **Introduction**
- Background & model
- Undirected vs directed communication
- Study of $s - t$ reachability
- Conclusion

Let's get Distributed



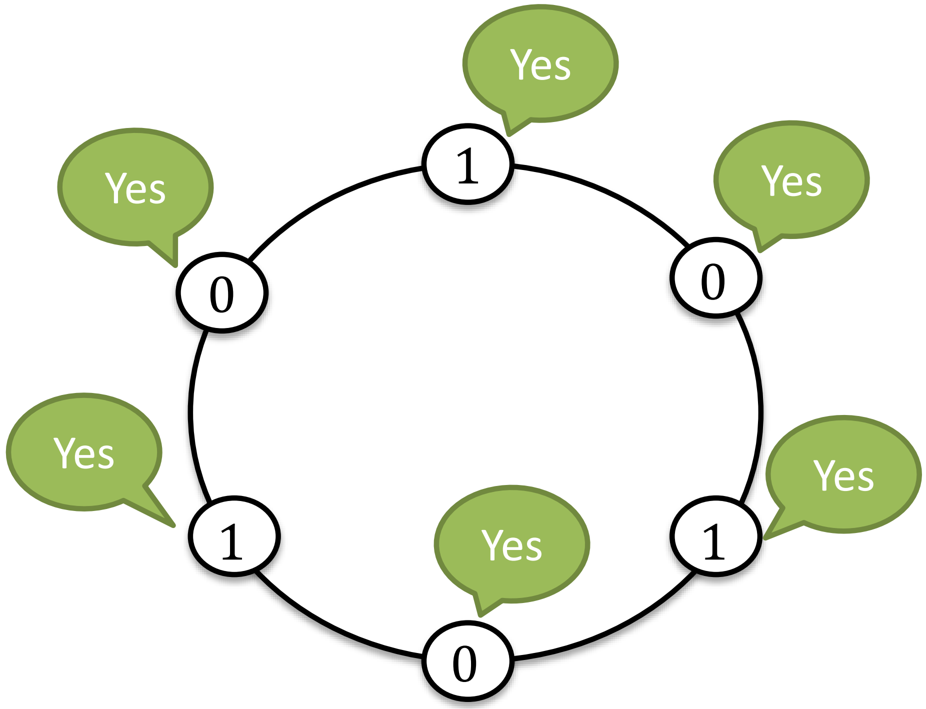
- Is n even?
- $\Omega(n)$ rounds, even with unique identifiers in the $\mathcal{LOCAL}$-model

Let's get Distributed



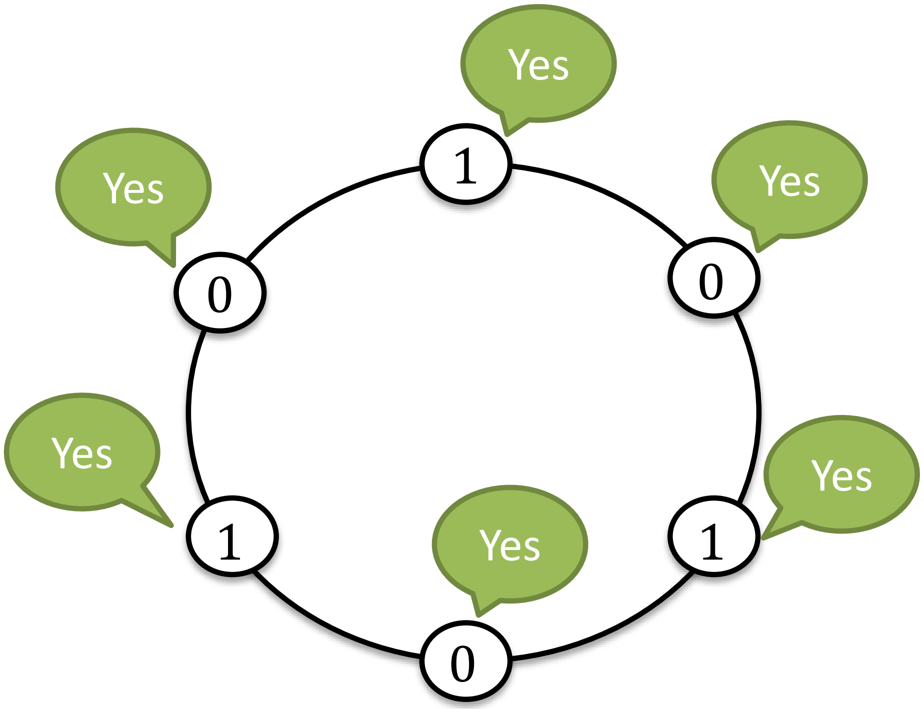
- Is n even?
- $\Omega(n)$ rounds, even with unique identifiers in the $\mathcal{LOCAL}$-model
- $\mathcal{P}$rover assigns 1 bit

Let's get Distributed



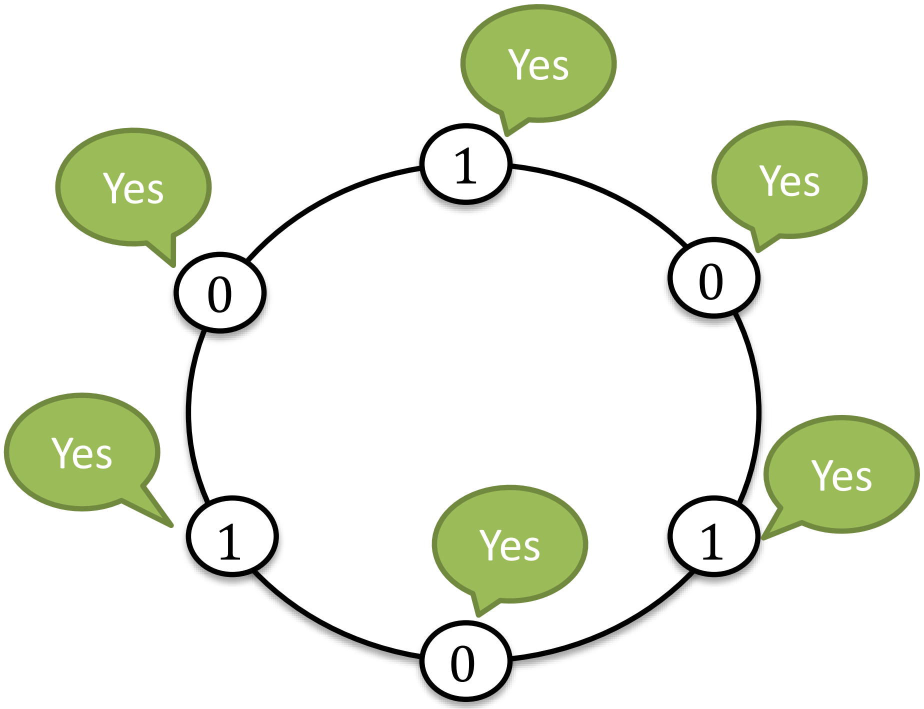
- Is n even?
- $\Theta(n)$ rounds in the $\mathcal{LOCAL}$-model
- $\mathcal{P}$rover assigns 1 bit -> $\mathcal{V}$erify in 1 round

Let's get Distributed



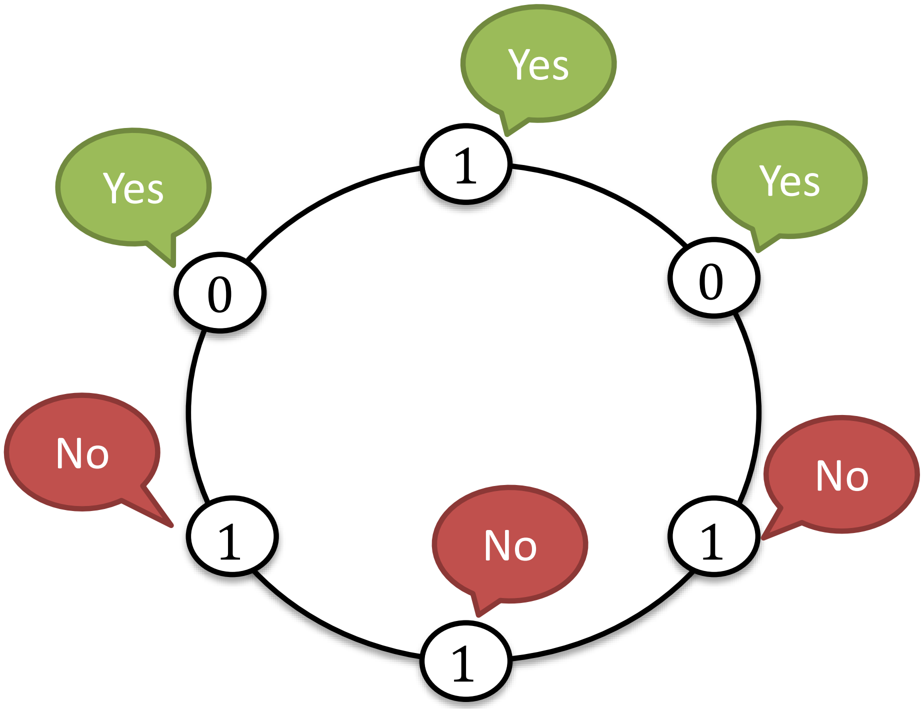
- Is n even?
- $\Theta(n)$ rounds in the $\mathcal{LOCAL}$-model
- $\mathcal{P}$rover assigns 1 bit -> $\mathcal{V}$erify in 1 round
- Other way to think of it: 1 bit of non-determinism
- General question: How many bits necessary/sufficient?

Accepting a proof



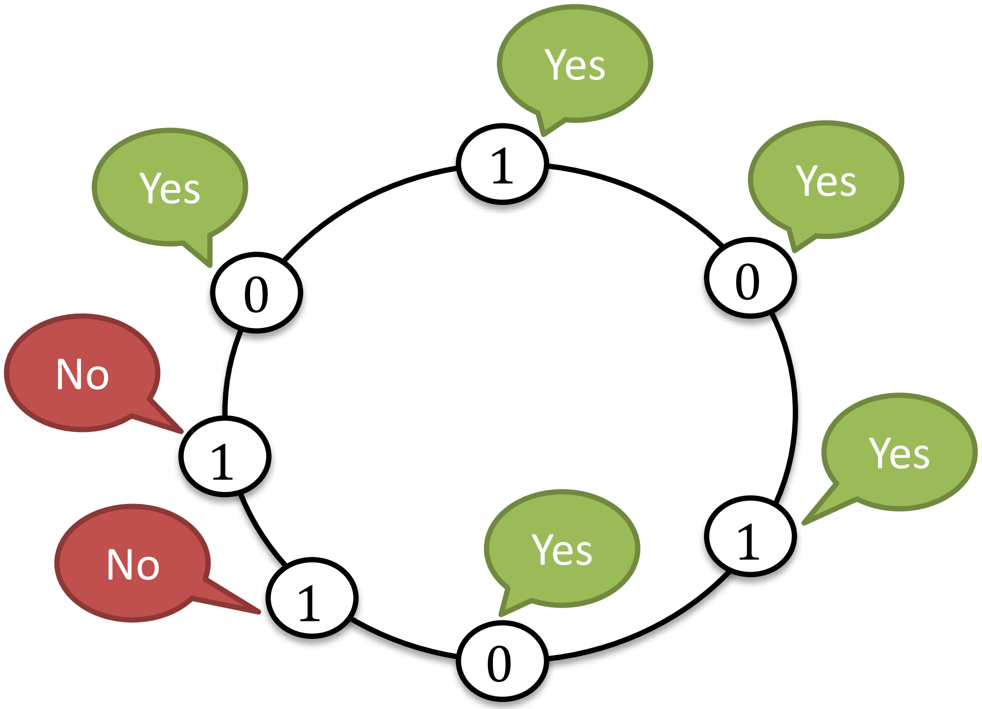
- Every node outputs **Yes** -> Proof accepted
- One node outputs **No** -> Proof rejected

Accepting a proof



- Every node outputs **Yes** -> Proof accepted
- One node outputs **No** -> Proof rejected
  - $\mathcal{P}$rover chose the wrong proof

Accepting a proof



- Every node outputs **Yes** -> Proof accepted
- One node outputs **No** -> Proof rejected
  - $\mathcal{P}$rover chose the wrong proof
  - Property does not hold

Overview

- Introduction
- Background & model
- Undirected vs directed communication
- Study of $s - t$ reachability
- Conclusion

Overview

- Introduction
- **Background & model**
- Undirected vs directed communication
- Study of $s - t$ reachability
- Conclusion

Some Related Work

- [Naor and Stockmeyer, STOC 1993]:
  What can be computed locally?
- [Göös and Suomela, PODC 2011]:
  Locally Checkable Proofs (LCP)
- [Korman et al., ICDCN 2006, …]:
  Proof Labeling Schemes (PLS)
- [Fraigniaud et al., FOCS 2011,…]:
  Nondeterministic Local Decision (NLD)
  – [Fraigniaud et al., DISC 2012,…]: “Randomization”

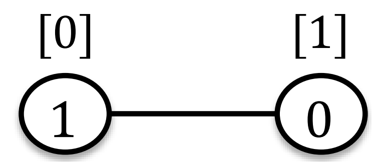
“No Strings attached”

- No knowledge of n
- No identifiers
- No port numbers
- No relaying of messages - just one round

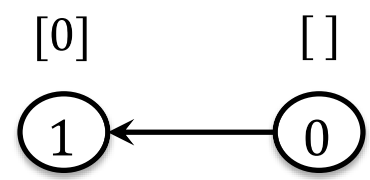
Graphs and Communication

- (Weakly) Connected graphs $G = (V, E)$ with $|V| = n$
  - **Yes** instances $\mathrm{G} \in \mathrm{Y}$ & **No** instances $\mathrm{G} \notin \mathrm{Y}$

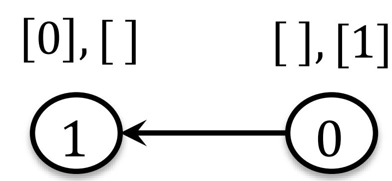
- Undirected: $\mathrm{U}(\mathrm{v})$ for every $\mathrm{v} \in V$
  - multiset of labels of all neighbors



- Directed: $\mathrm{D}_1(\mathrm{v})$ for every $\mathrm{v} \in V$
  - Multiset I of labels of all incoming-neighbors



- Directed: $\mathrm{D}_2(\mathrm{v})$ for every $\mathrm{v} \in V$
  - two multisets (I,O) of labels of all
    - incoming-neighbors
    - outgoing-neighbors



Local Checkability

- $\mathcal{P}$rover $\mathcal{P}$ gets as input $\mathrm{G} \in \mathrm{Y}$
  - Assigns a labels $\ell(\mathrm{v})$ for every $\mathrm{v} \in V$

- $\mathcal{V}$erifier $\mathcal{V}$ is a distributed algorithm that gets as input at node v both $\ell(\mathrm{v})$ & $\mathrm{U}(\mathrm{v})$ (or $\mathrm{D}_1(\mathrm{v})$ / $\mathrm{D}_2(\mathrm{v})$)
  - Outputs either **Yes** or **No**

- A Prover-Verifier pair ($\mathcal{P}$,$\mathcal{V}$) is correct for Y if:
  - $\mathrm{G} \in \mathrm{Y}$ & labels from $\mathcal{P}$: $\mathcal{V}$ outputs **Yes** at all nodes
  - $\mathrm{G} \notin \mathrm{Y}$: $\mathcal{V}$ outputs **No** for at least one node

Prover-Verifier Pairs

- We investigate if there are correct ($\mathcal{P}$,$\mathcal{V}$) for some Y
  - (abbreviated by U-PVP, D_1-PVP, D_2-PVP)

- The quality of a PVP is its *proof size*
  - $f(n)$, if the PVP uses at most $f(n)$ bits for each label in any **Yes** instance with at most n nodes

- The U-*proof size* of Y is the smallest proof size for which there exists a correct U-PVP
  - Analogous for D_1-proof size / D_2-proof size

- In this talk: All logarithms are of base 2 and rounded up to be of integer value

Overview

- Introduction
- **Background & model**
- Undirected vs directed communication
- Study of $s - t$ reachability
- Outlook

Overview

- Introduction
- Background & model
- **Undirected vs directed communication**
- Study of $s - t$ reachability
- Outlook

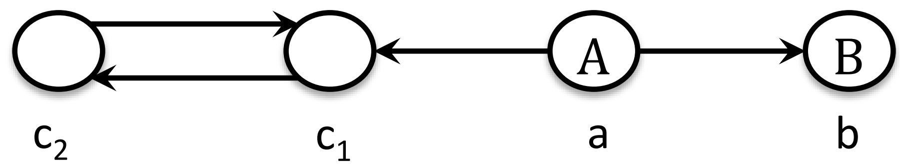
Undirected vs Directed Communication

- The different models can induce different amount of bits required in the proof size
  - Or might even render a problem impossible

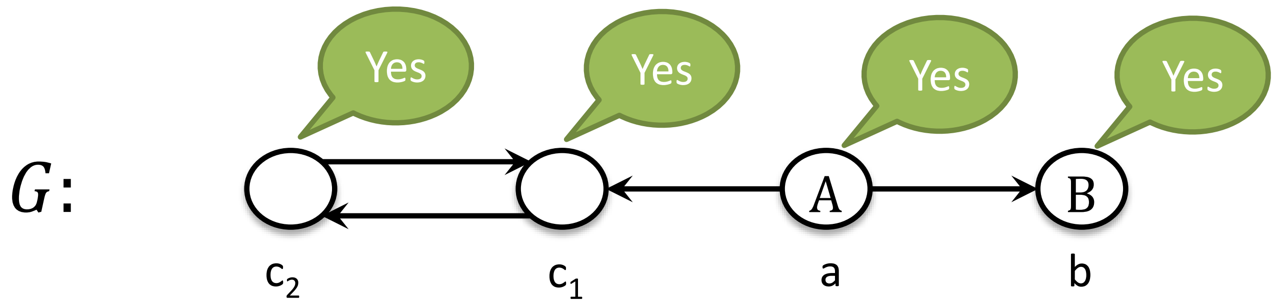
- Example problem Υ : *CYCLE*
  - *U-CYCLE*: all undirected graphs containing a cycle
  - *D-CYCLE*: all directed graphs containing a directed cycle

D-CYCLE: Is there a D_1-PVP?

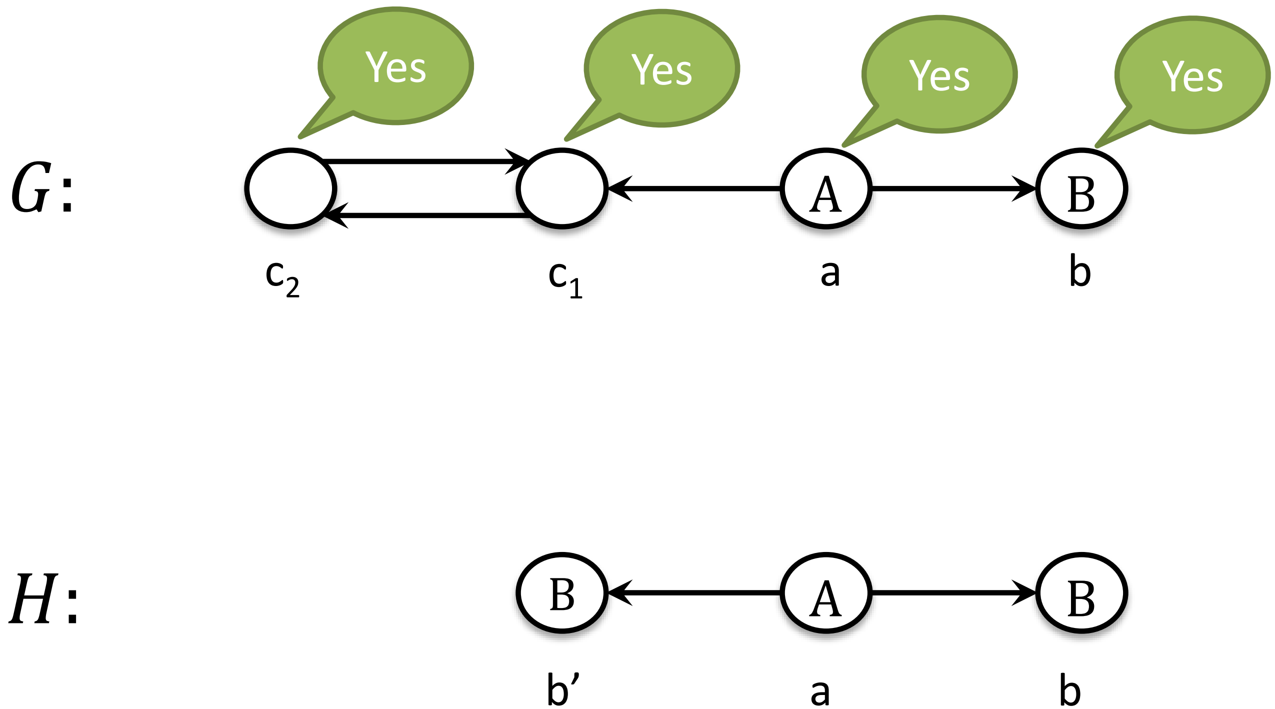
G:



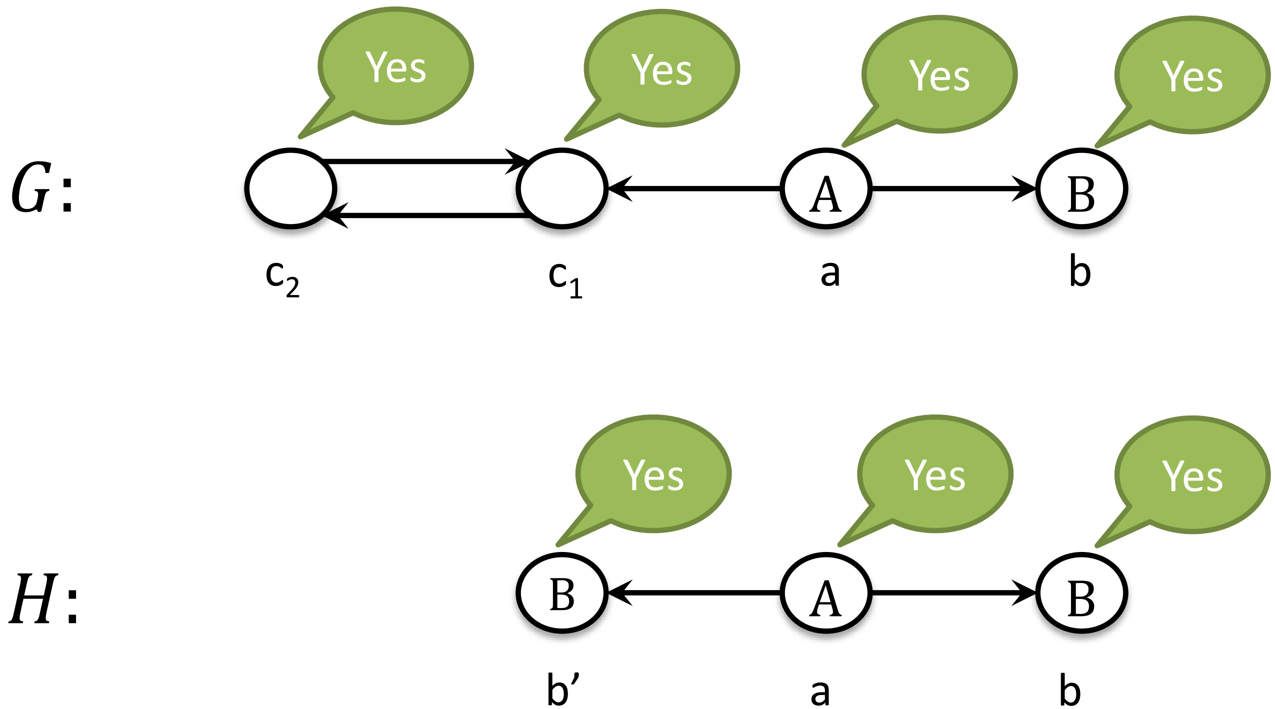
D-CYCLE: Is there a D_1-PVP?



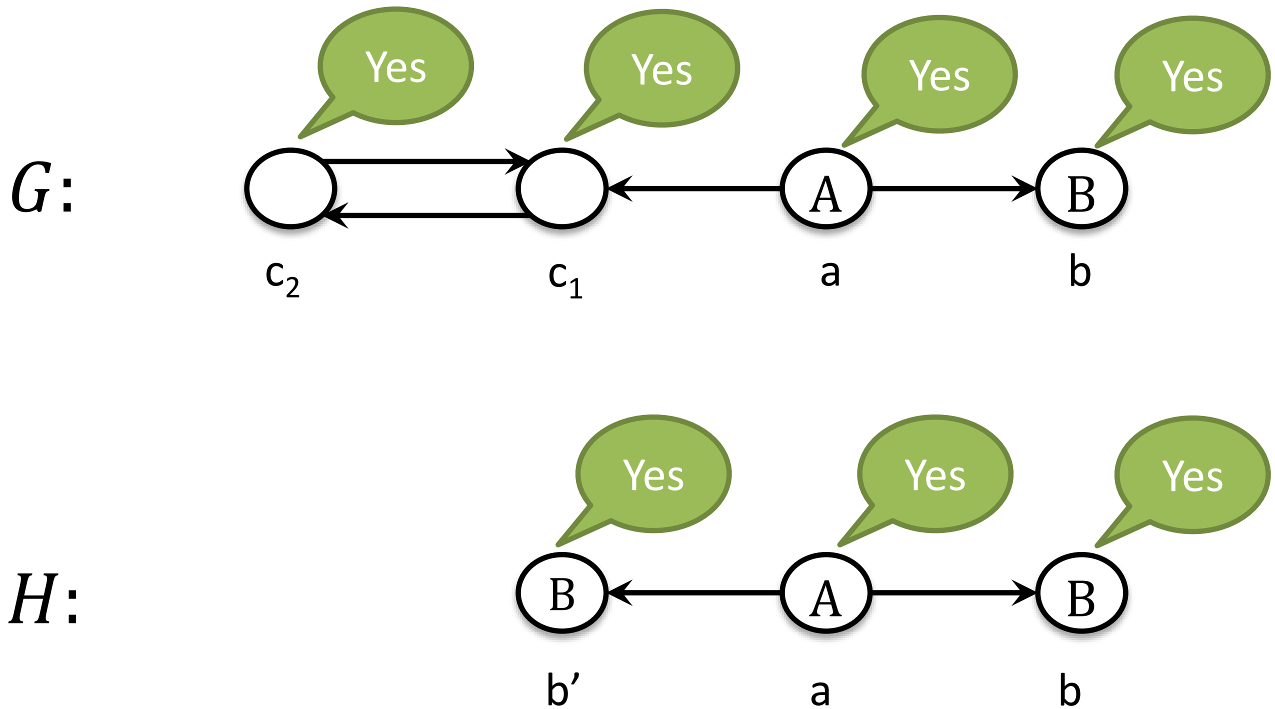
D-CYCLE: Is there a D_1-PVP?



D-CYCLE: Is there a D_1-PVP?



D-CYCLE: Is there a D_1-PVP?



There is no D_1-PVP for D-CYCLE

CYCLE

Problem	Directed one-way	Directed two-way	Undirected
CYCLE	Impossible		

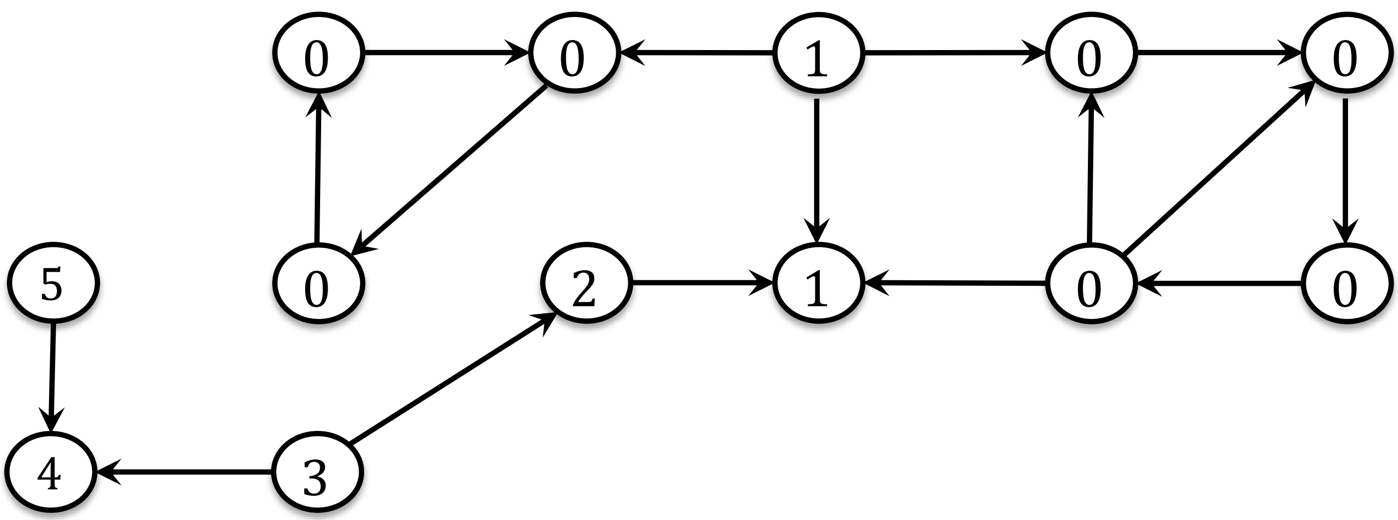
D-CYCLE: Is there a D_2-PVP?

D-CYCLE: Is there a D_2-PVP?

- $\mathcal{P}$rover $\mathcal{P}$ labels nodes as follows:
  - In a directed cycle? -> 0
  - Else: Minimum distance to a cycle
    - (in the underlying undirected graph)
  - Proof size: $\log n$ bits

D-CYCLE: Is there a D_2-PVP?

- $\mathcal{P}$rover $\mathcal{P}$ labels nodes as follows:
  - In a directed cycle? -> 0
  - Else: Minimum distance to a cycle
    - (in the underlying undirected graph)
  - Proof size: $\log n$ bits

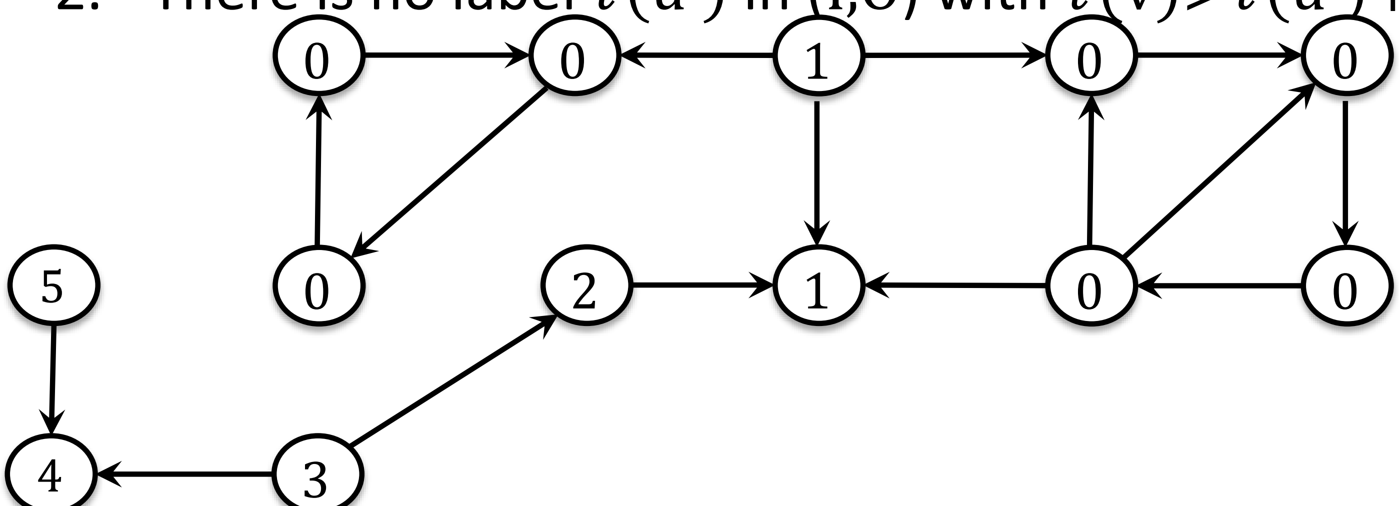


D-CYCLE: Is there a D_2-PVP?

- $\mathcal{V}$erifier $\mathcal{V}$ returns **Yes**
  - For nodes $\mathrm{v_c}$ with label $\ell(\mathrm{v_C})=0$ if for (I,O) holds:
    - $0 \in \mathrm{O}$ and $0 \in \mathrm{I}$
  - For the other nodes v with label $\ell(\mathrm{v})$ if
    1. There is a label $\ell(\mathrm{u})$ in (I,O) with $\ell(\mathrm{v})=\ell(\mathrm{u})+1$, and
    2. There is no label $\ell(\mathrm{u'})$ in (I,O) with $\ell(\mathrm{v})>\ell(\mathrm{u'})+1$

D-CYCLE: Is there a D_2-PVP?

- $\mathcal{V}$erifier $\mathcal{V}$ returns **Yes**
  - For nodes $\mathrm{v_c}$ with label $\ell(\mathrm{v_C})=0$ if for (I,O) holds:
    - $0 \in \mathrm{O}$ and $0 \in \mathrm{I}$
  - For the other nodes v with label $\ell(\mathrm{v})$ if
    1. There is a label $\ell(\mathrm{u})$ in (I,O) with $\ell(\mathrm{v})=\ell(\mathrm{u})+1$, and
    2. There is no label $\ell(\mathrm{u}')$ in (I,O) with $\ell(\mathrm{v})>\ell(\mathrm{u}')+1$

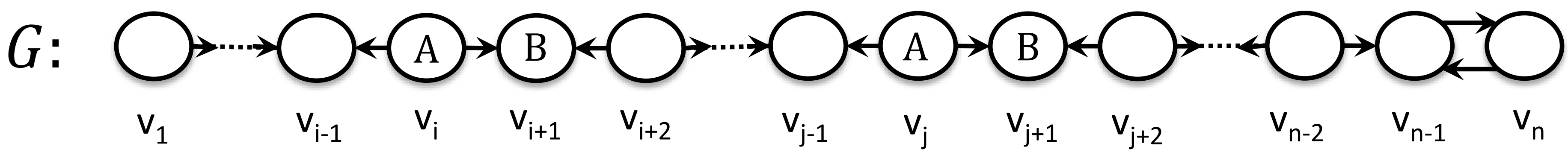


Is the described D_2-PVP correct?

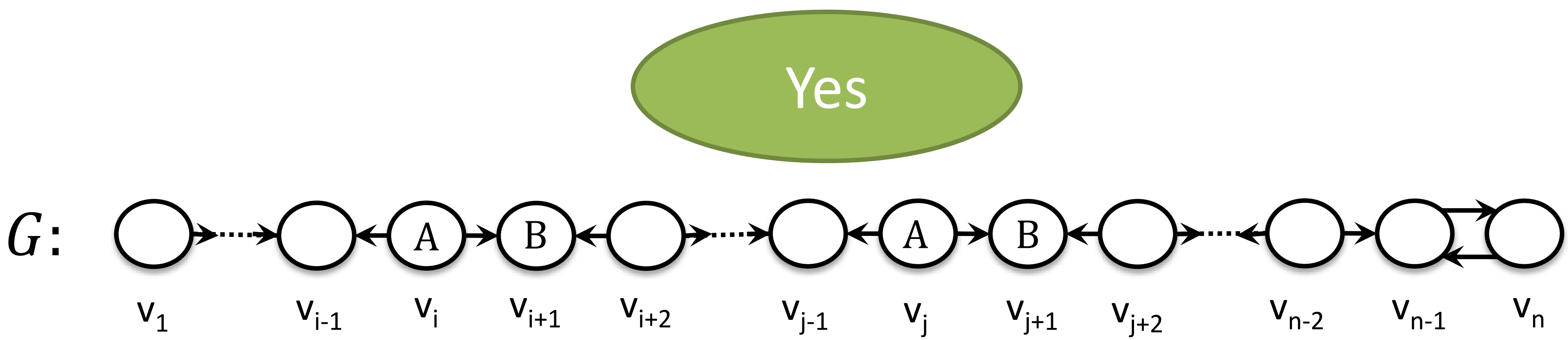
- **Yes** instances labeled by $\mathcal{P}$:
  - Only nodes in directed cycles labeled with 0 -> **Yes**
  - All other nodes: Label is defined by minimum distance to a directed cycle -> **Yes**

- **No** instances:
  - Is there a node with label 0? Follow “0-path” -> **No**
  - No node with label 0, but one with label k?
    - Follow “descending path” -> **No**

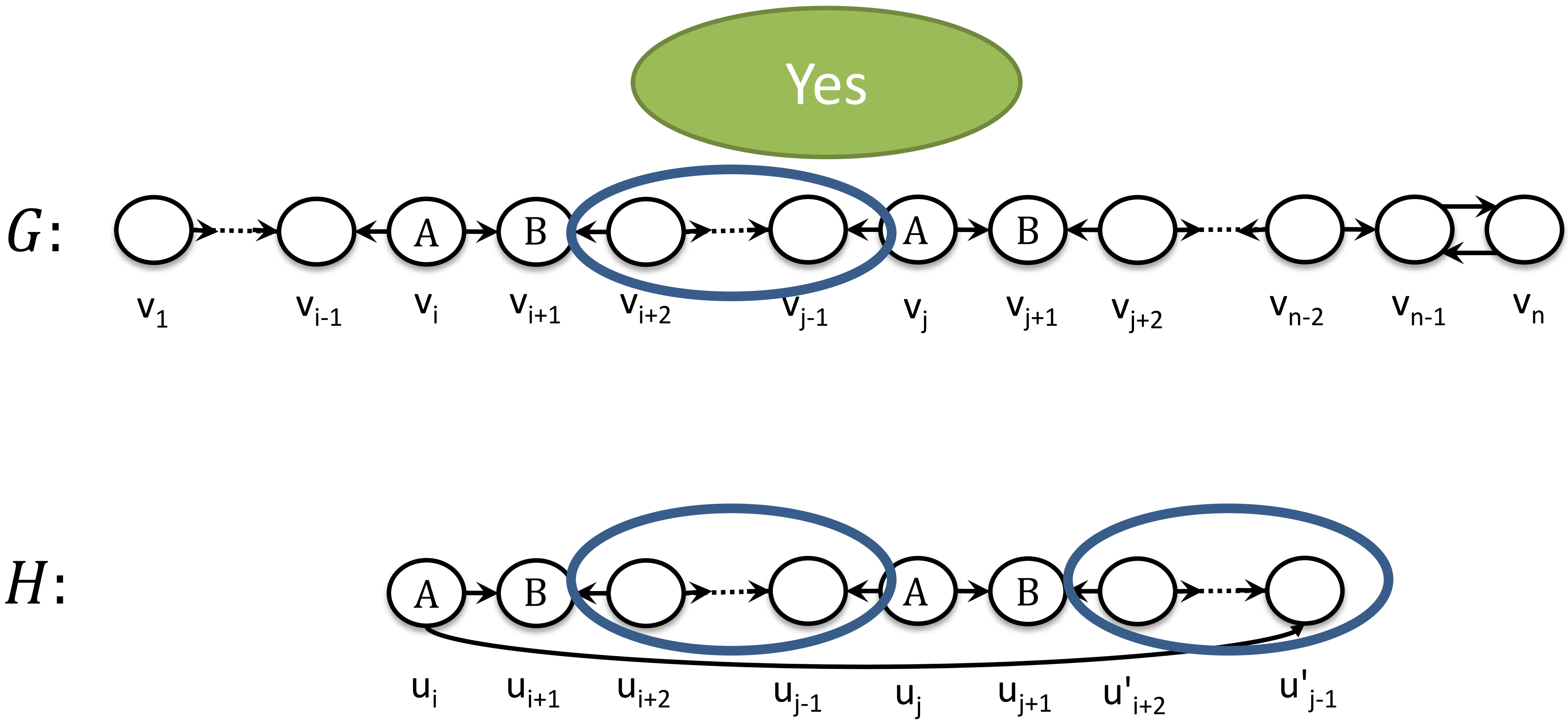
D_2-proof size: $\Omega(\log n)$ bits



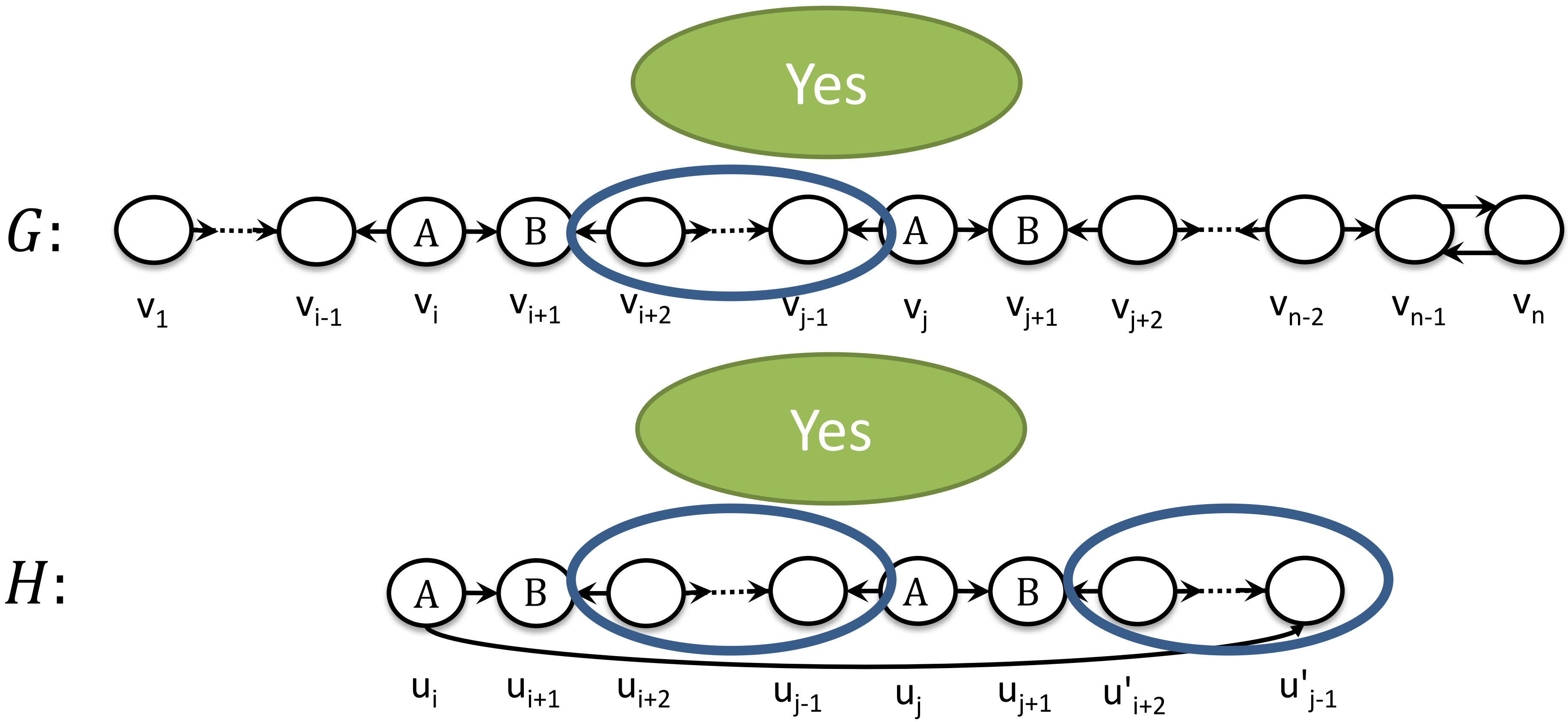
D_2-proof size: $\Omega(\log n)$ bits



D_2-proof size: $\Omega(\log n)$ bits



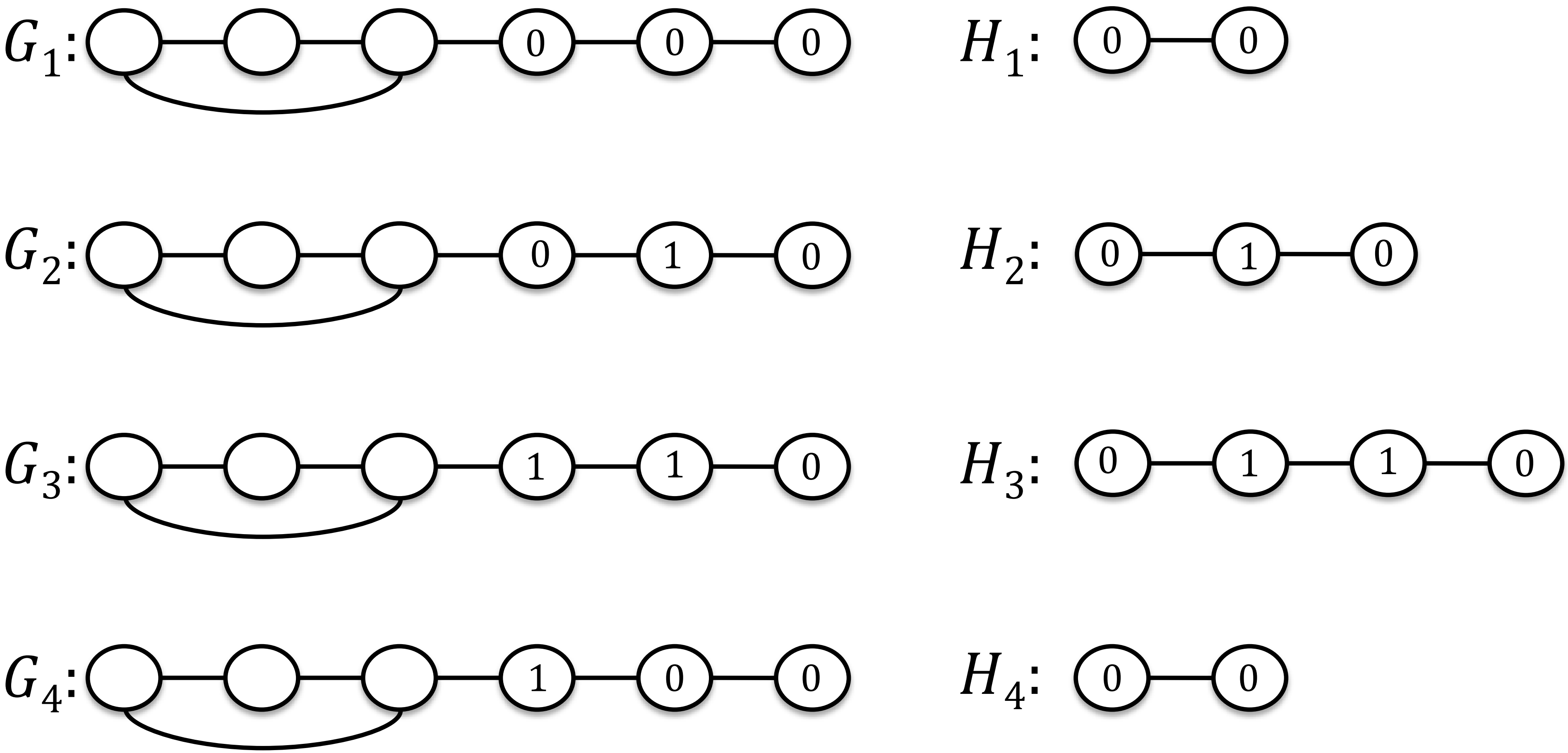
D_2-proof size: $\Omega(\log n)$ bits



CYCLE

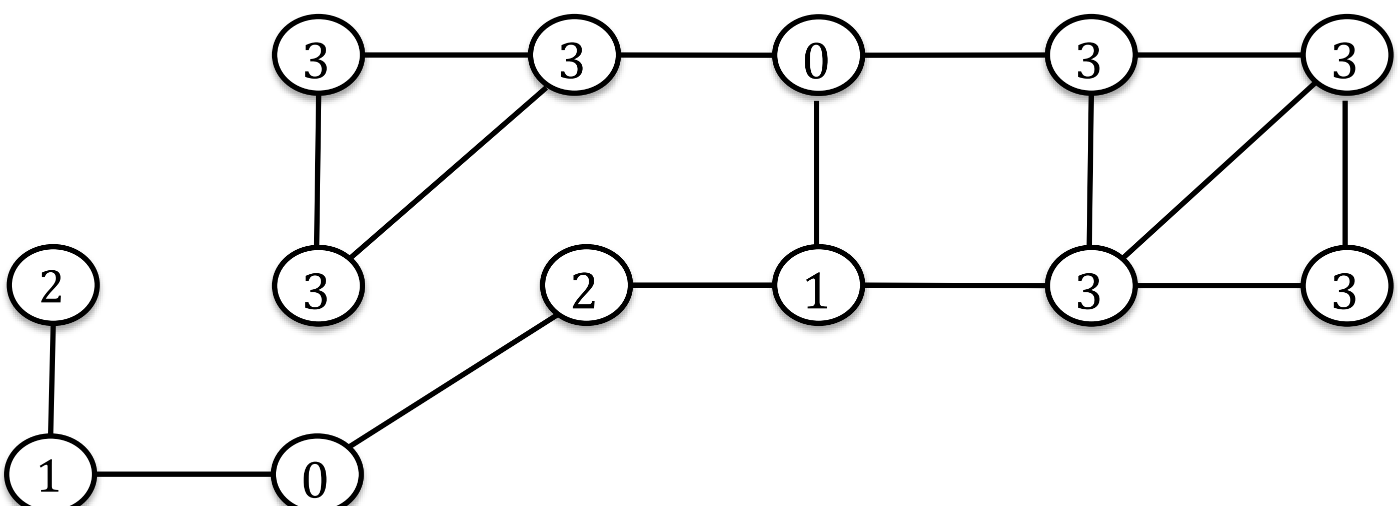
Problem	Directed one-way	Directed two-way	Undirected
CYCLE	Impossible	$\Theta(\log n)$	

U-proof size: At least 2 Bits



U-PVP for CYCLE with 2 bits

- $\mathcal{P}$rover $\mathcal{P}$ labels nodes as follows:
  - In a cycle? -> 3
  - Else: Remove all cycles, remaining graph is a forest
    - For each tree T:
      - » Create a root r adjacent to a cycle in G with label 0
      - » Other nodes: Distance to r modulo 3
  - Proof size: 2 bits

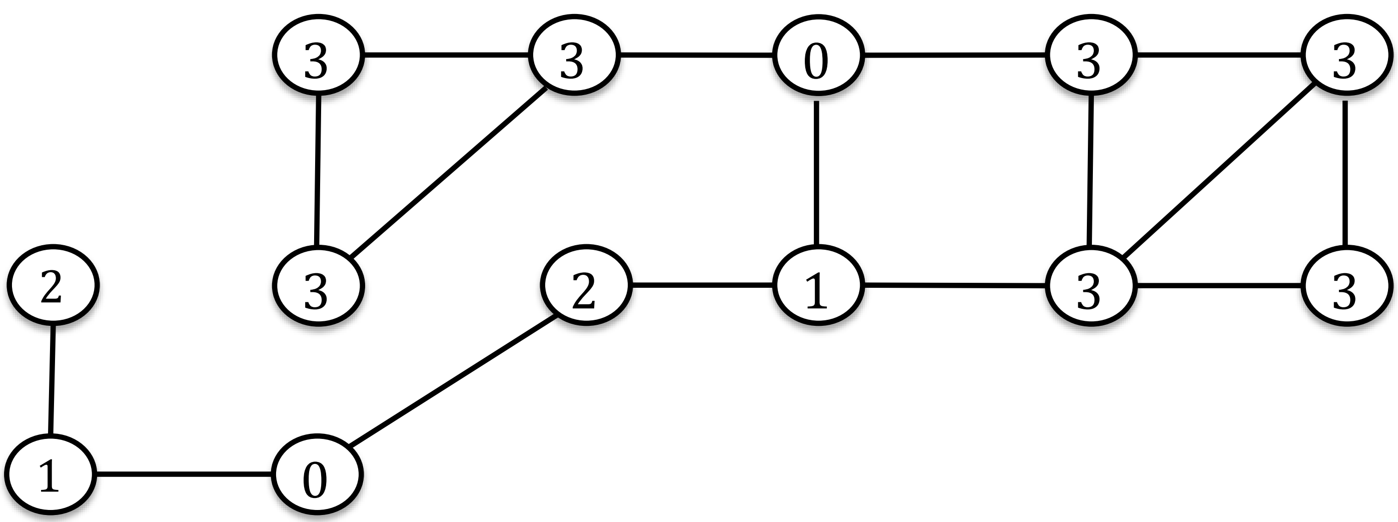


U-PVP for CYCLE with 2 bits

- $\mathcal{V}$erifier $\mathcal{V}$ returns **Yes**
  - For nodes v_{c} with label $\ell(\mathrm{v}_{\mathrm{C}})=3$ if holds:
    - Two neighbors with label 3 exist
  - For the other nodes v with label $\ell(\mathrm{v}) \in \{0,1,2\}$ if
    1. There is no neighbor with label $\ell(\mathrm{v})$, and
    2. Exactly one neighbor exists with label $\ell(\mathrm{v})-1$ mod 3 or at least one neighbor with label of 3

Is the described U-PVP correct?

- **Yes** instances labeled by $\mathcal{P}$:
  - Only nodes in cycles labeled with 3 -> **Yes**
  - Without the cycles, all other nodes are in a tree with labels as distance to root mod 3, and root is adjacent to a cycle -> **Yes**



Is the described U-PVP correct?

- **Yes** instances labeled by $\mathcal{P}$:
  - Only nodes in cycles labeled with 3 -> **Yes**
  - Without the cycles, all other nodes are in a tree with labels as distance to root mod 3, and root is adjacent to a cycle -> **Yes**

- **No** instances (without a cycle):
  - Is there a node with label 3? They form a forest, consider any leaf-> **No**
  - Else: follow “descending path” -> **No**

CYCLE, ACYCLIC, TREE

Problem	Directed one-way	Directed two-way	Undirected
CYCLE	Impossible	$\Theta(\log n)$	2

CYCLE, ACYCLIC, TREE

Problem	Directed one-way	Directed two-way	Undirected
CYCLE	Impossible	$\Theta(\log n)$	2
TREE	$\Theta(\log n)$*	$\Theta(\log n)$	$\Theta(\log n)$*
ACYCLIC	$\Theta(\log n)$	$\Theta(\log n)$	same as Tree

Idea for Tree:

- Label root as 0
- Other nodes: Label is distance from root

Idea for Acyclicity:

- Label nodes without incoming edges as 0
- Other nodes: Max. incoming label plus 1

*: [Korman et al., Distributed Computing 2010]: Proof labeling schemes

Overview

- Introduction
- Background & model
- **Undirected vs directed communication**
- Study of $s - t$ reachability
- Conclusion

Overview

- Introduction
- Background & model
- Undirected vs directed communication
- **Study of $s - t$ reachability**
- Conclusion

$s - t$ Reachability

- Is there a (directed) path from s to t?

“To ask meaningful questions about connectivity […] we have the promise that there is exactly one node with label s and exactly one node with label t.“

[Göös and Suomela, PODC 2011]

- We thus assume that there are two nodes with the unique labels s and t

- U-proof size of 1 bit (e.g., [Immermann, 1999]):
  - Label nodes along a shortest $s - t$ path with 1, else 0

Directed $s - t$ Reachability

- D_2-PVP with port numbers: $O(\log \Delta)$ bits
  - With Δ being max degree
  - Idea: "Point at successor and predecessor" along a shortest $s - t$ path
- Open question:

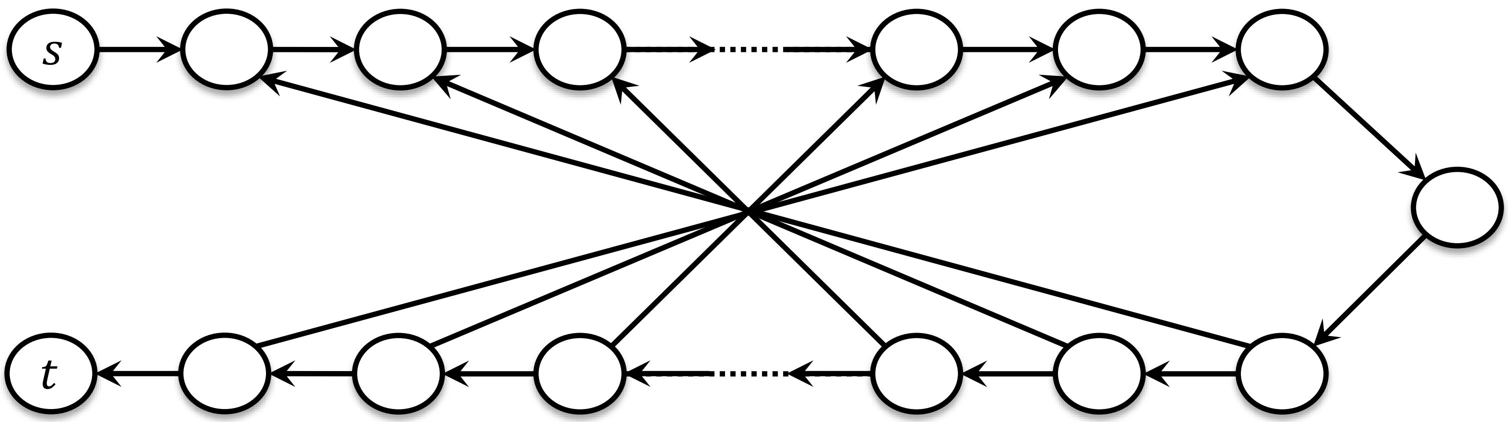
"Is there a proof labelling scheme with O(1)-bit proofs?"

[Göös and Suomela, PODC 2011]

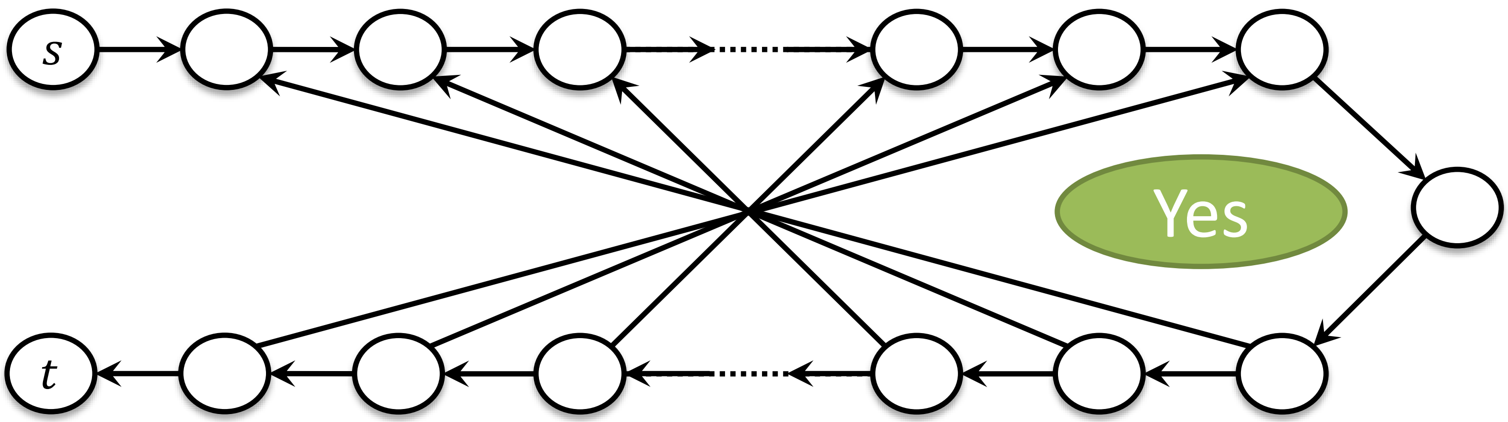
D_1-PVP for $s - t$ Reachability

- We don't have port numbers…
- Idea: Take a shortest $s - t$ path $s, \mathrm{v}_1, \ldots \mathrm{vj}, t$
  - Label according to distance to s along the path
  - All other nodes: Label of 0
- Proof size of $\log n$

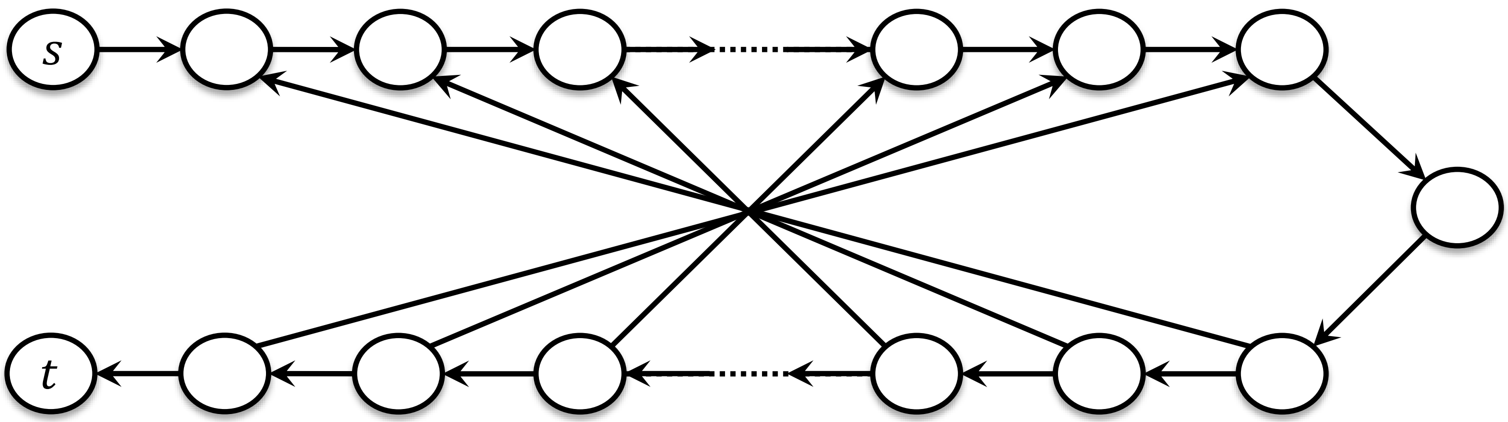
D_1-proof size: $\Omega(\log n)$ bits



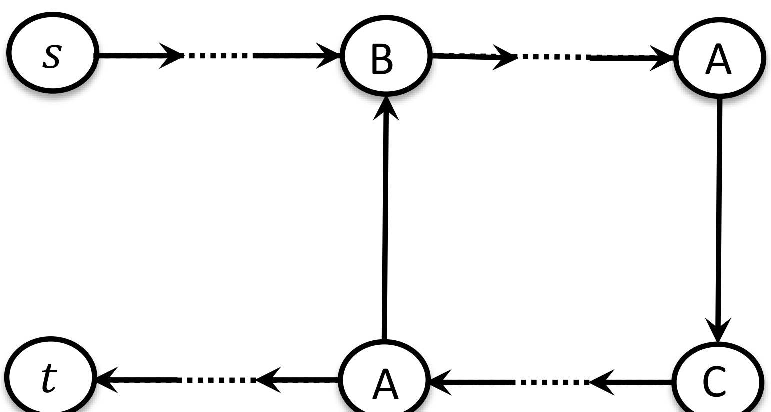
D_1-proof size: $\Omega(\log n)$ bits



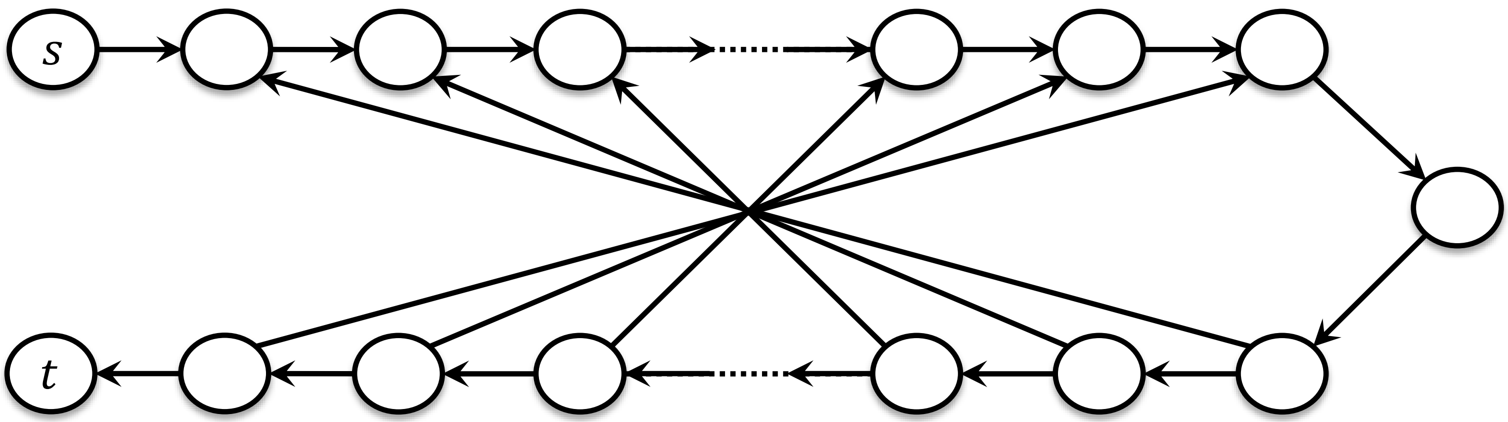
D_1-proof size: $\Omega(\log n)$ bits



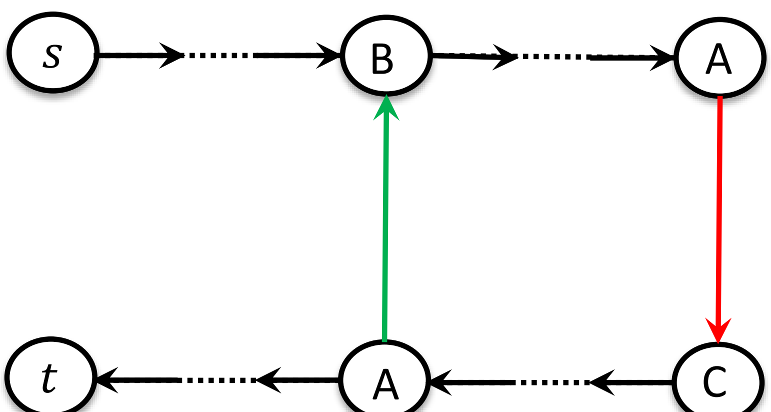
G:



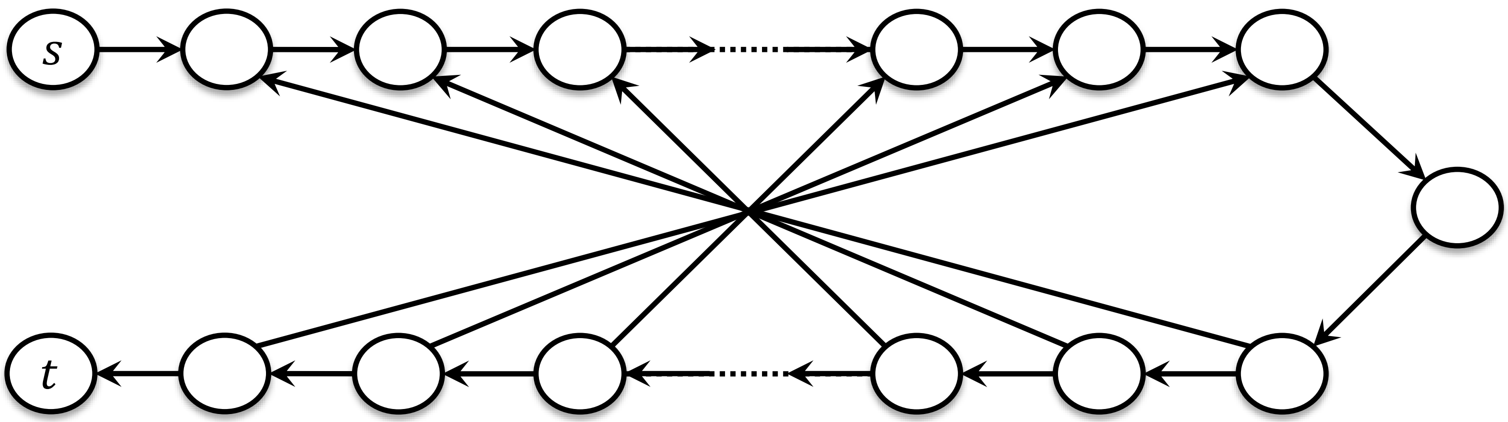
D_1-proof size: $\Omega(\log n)$ bits

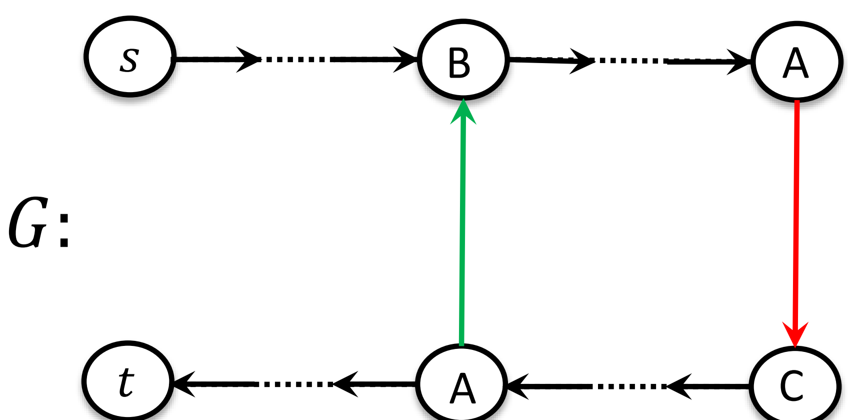


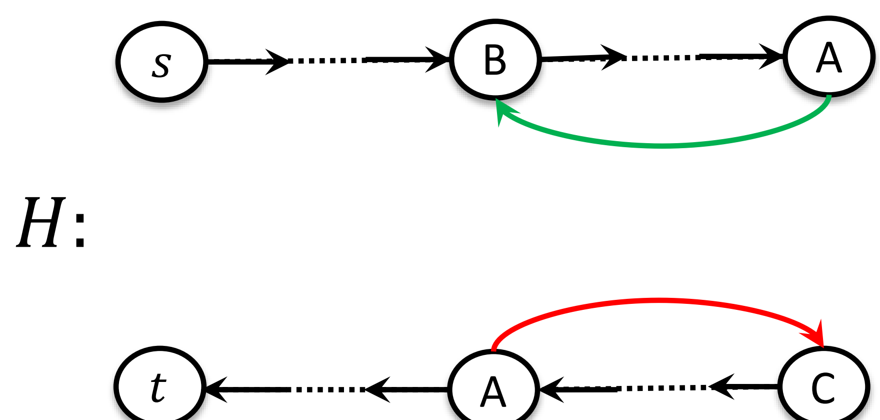
G:



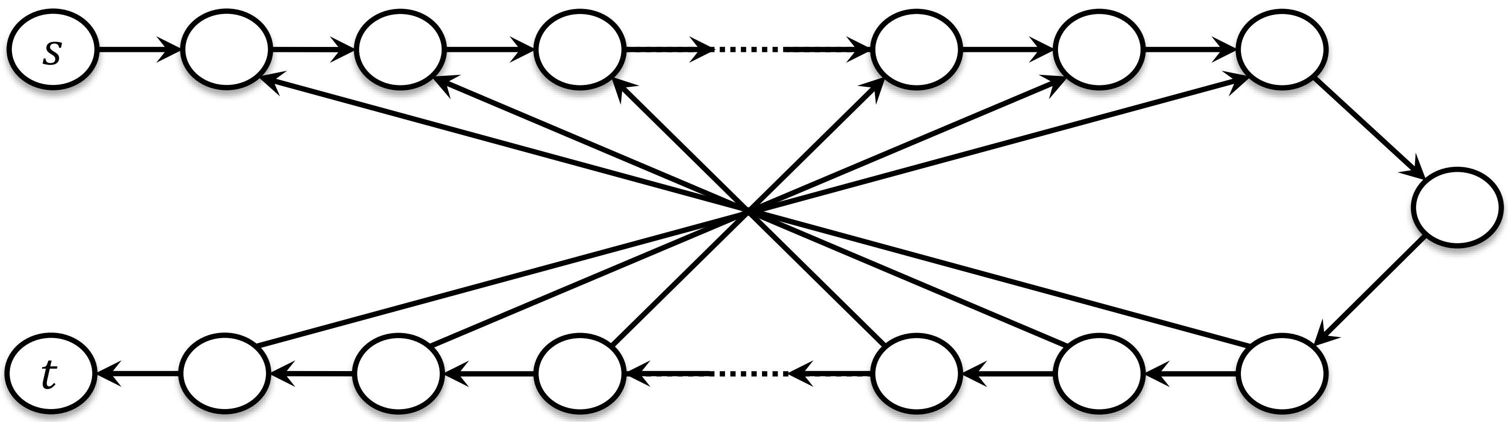
D_1-proof size: $\Omega(\log n)$ bits



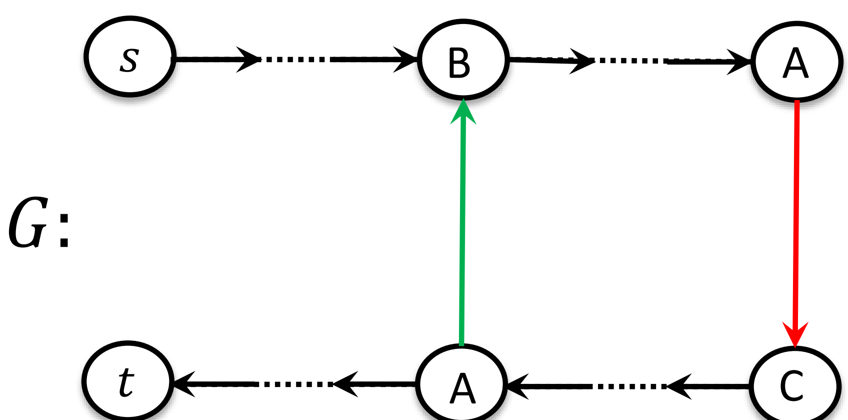


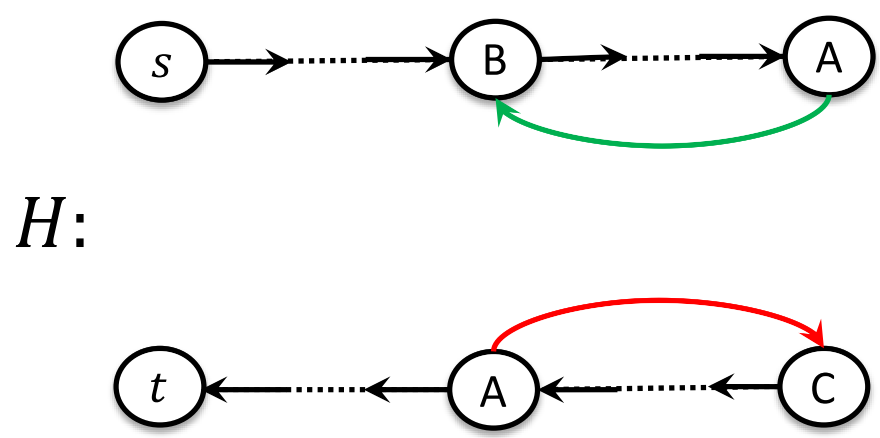


D_1-proof size: $\Omega(\log n)$ bits



There is no D_1-PVP with $f(\Delta)$ bits!





D_2-PVP for $s - t$ Reachability

- As we don't have port numbers, we could use the D_1-PVP with $\log n$ bits

- With port numbers: $O(\log \Delta)$ bits

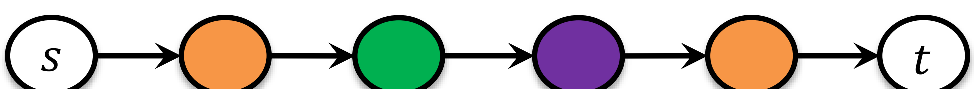
- Let us create port numbers!

D_2-PVP for $s - t$ Reachability

- Idea: A 2-hop coloring needs $\leq \Delta^2+1$ colors
  - Encoding each color: $O(\log \Delta)$ bits

- 2-hop coloring can be checked locally
  - All colors in the 1-hop neighborhood different?

- Thus, we can point "back and forth" along edges, by emulating port numbers with $O(\log \Delta)$ bits



Conclusion

- Summary
  - All three models of communication differ
  - Our lower bound examples have constant degree
    - Can drop the 1 round restriction and go local
  - Directed $s - t$ reachability:
    - One-Way: Proof size of $\Theta(\log n)$ bits, $f(\Delta)$ bits don't suffice
    - Two-Way: Emulating port numbers -> $O(\log \Delta)$ bits proof size

- Open Questions
  - What happens in biologically inspired systems?
    - E.g., no multisets but sets & finite automata verifier?
  - What is the correct answer to D_2 $s - t$ reachability?
  - Can similar techniques be deployed in production networks?

Thank you

__Klaus-Tycho Förster__, Thomas Lüdi, Jochen Seidel, Roger Wattenhofer
January 06, 2016 @ ICDCN 2016 - Singapore