Distributed Online Data Aggregation in Dynamic Graphs

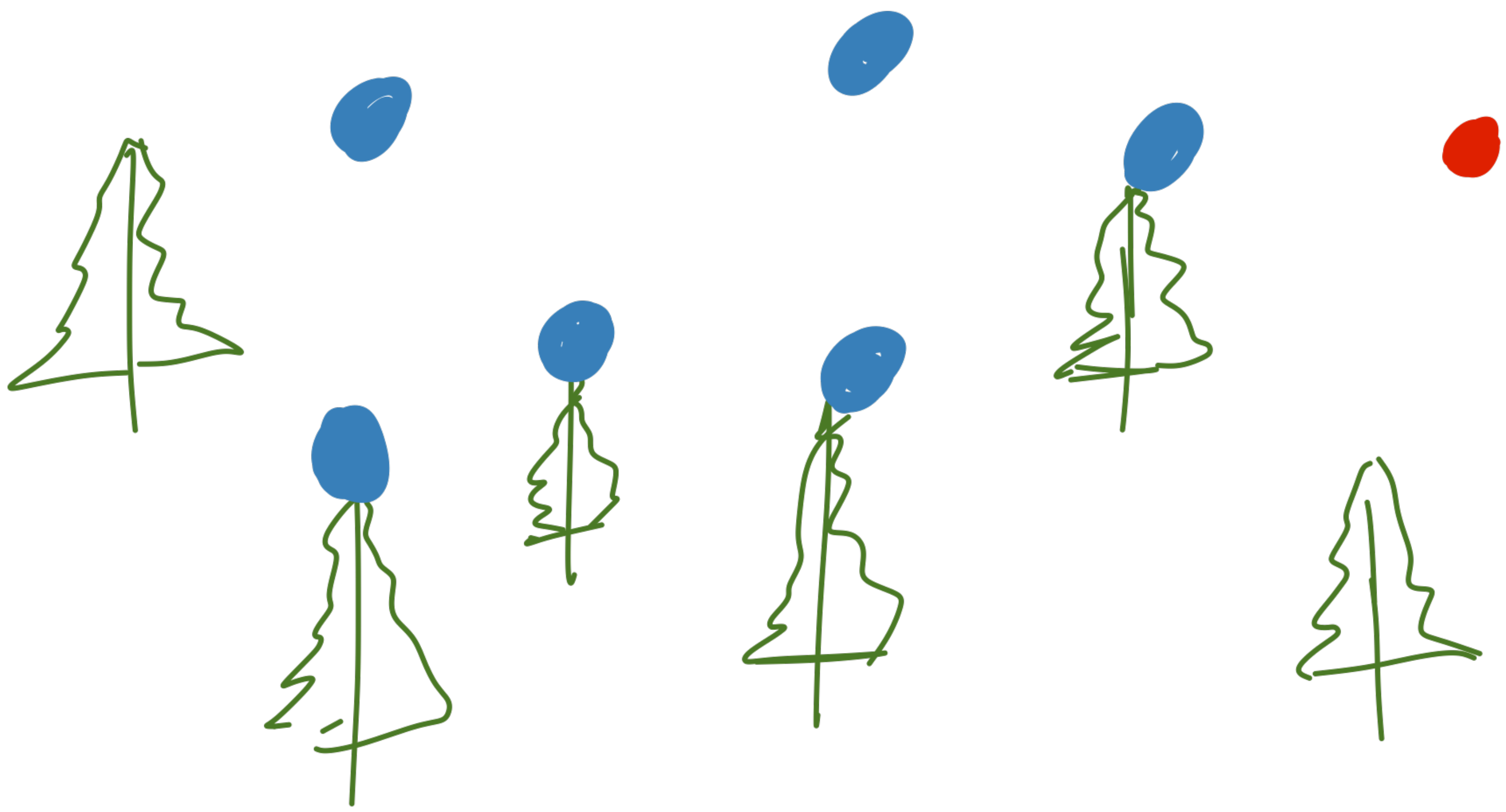
Quentin Bramas, Toshimitsu Masuzawa, and Sébastien Tixeuil

NETYS 2019, Marrakech, June, 21st

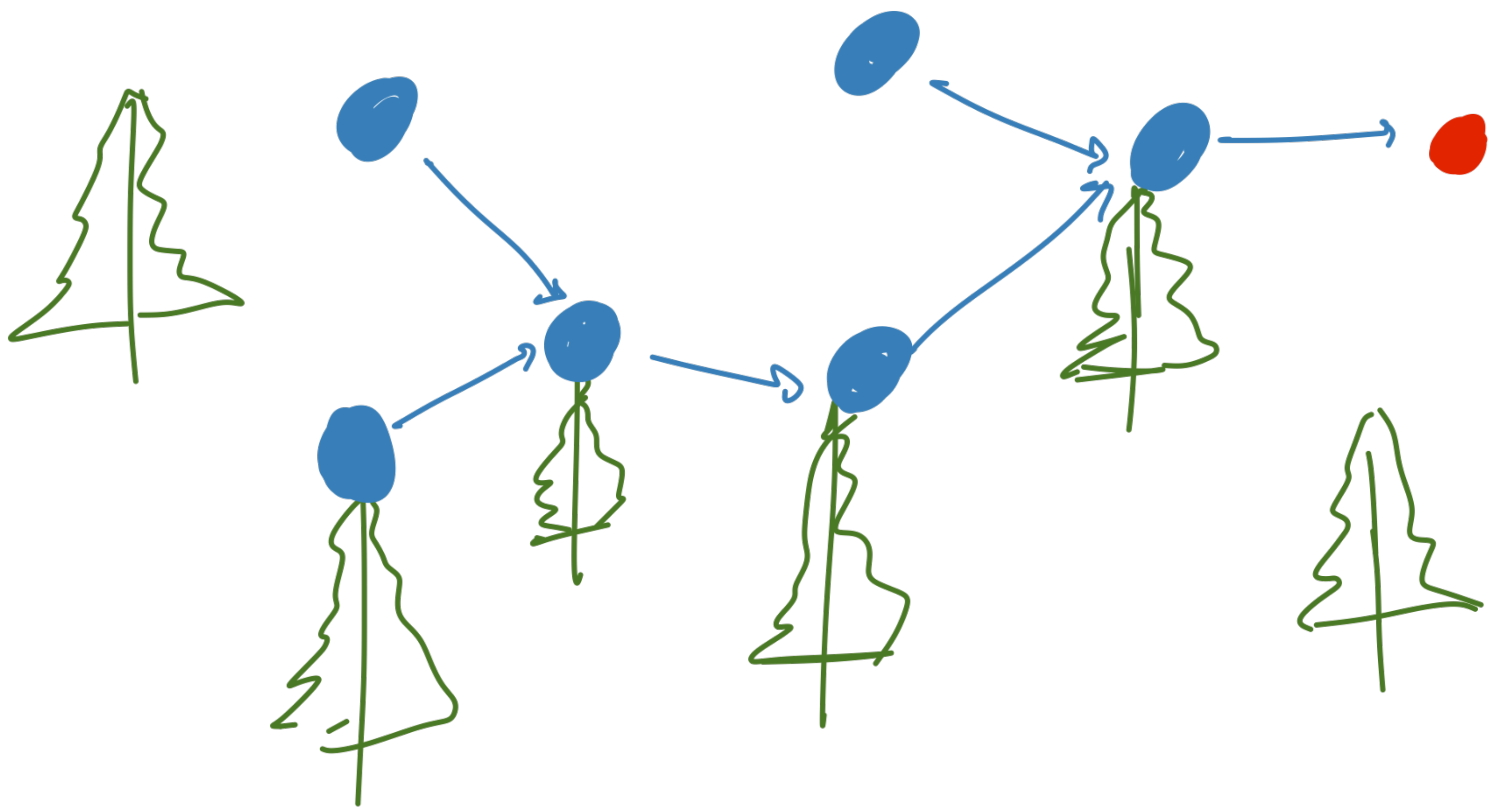
bramas@unistra.fr

Data Aggregation Problem

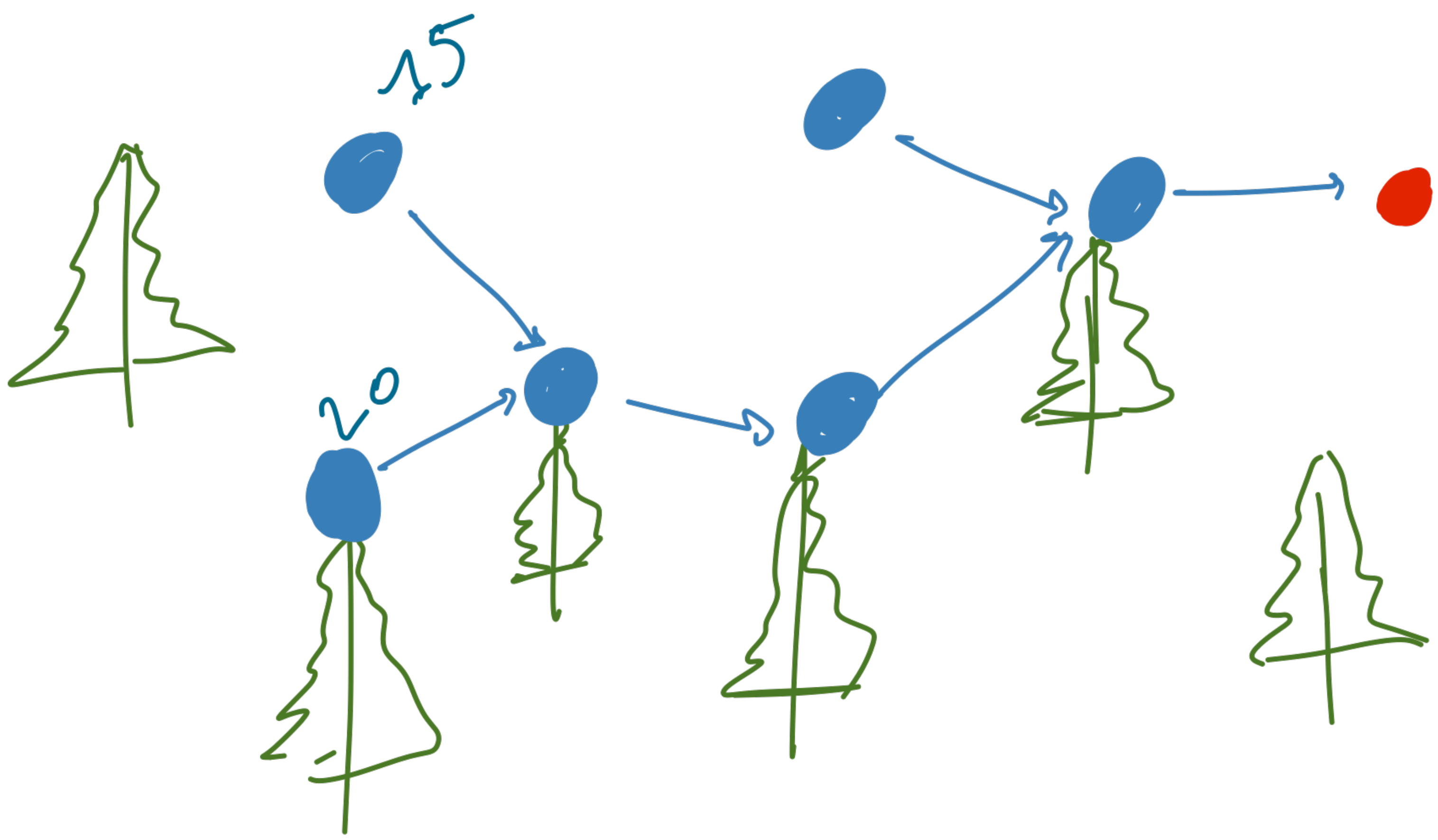
Data Aggregation Problem



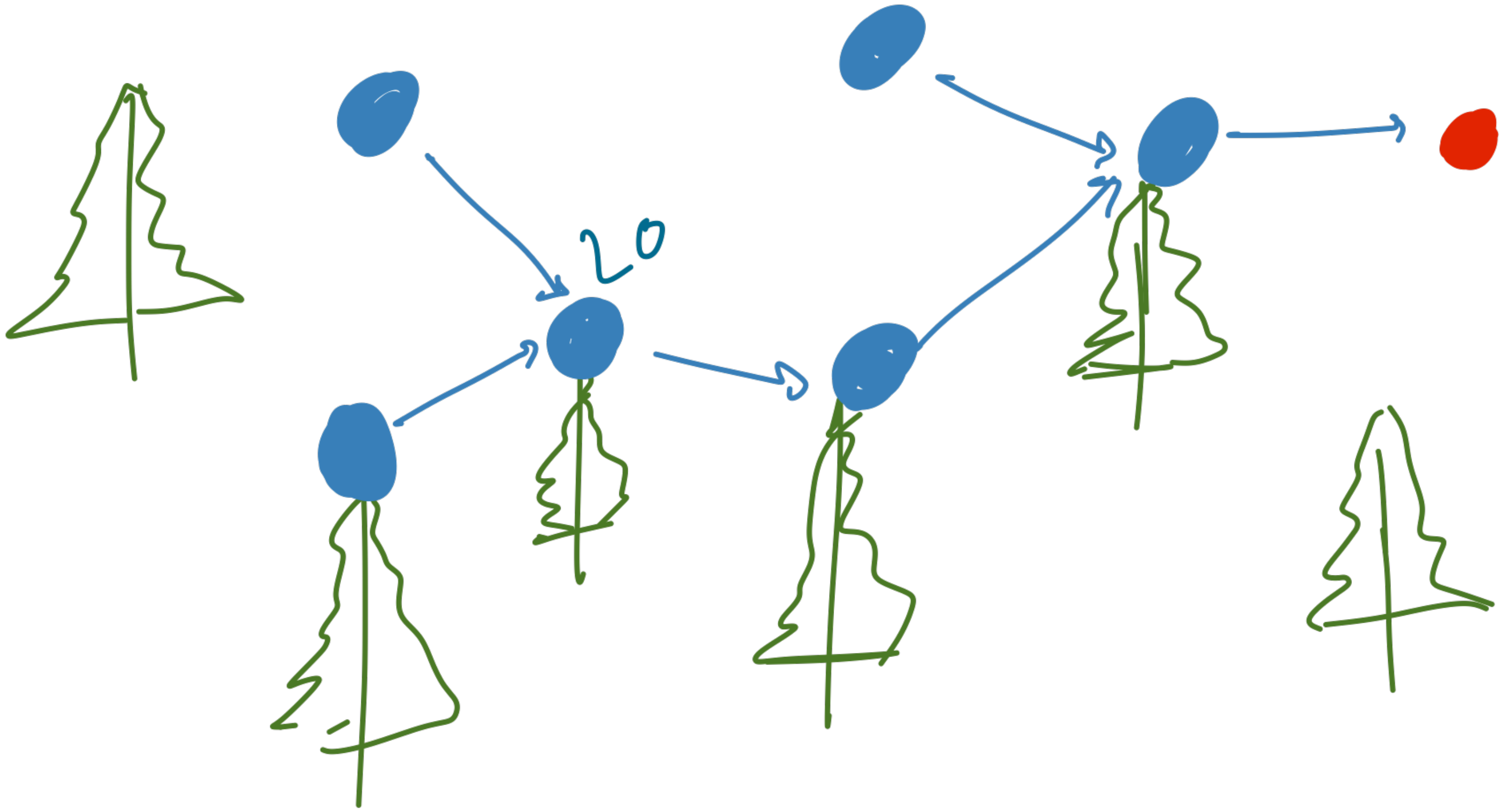
Data Aggregation Problem



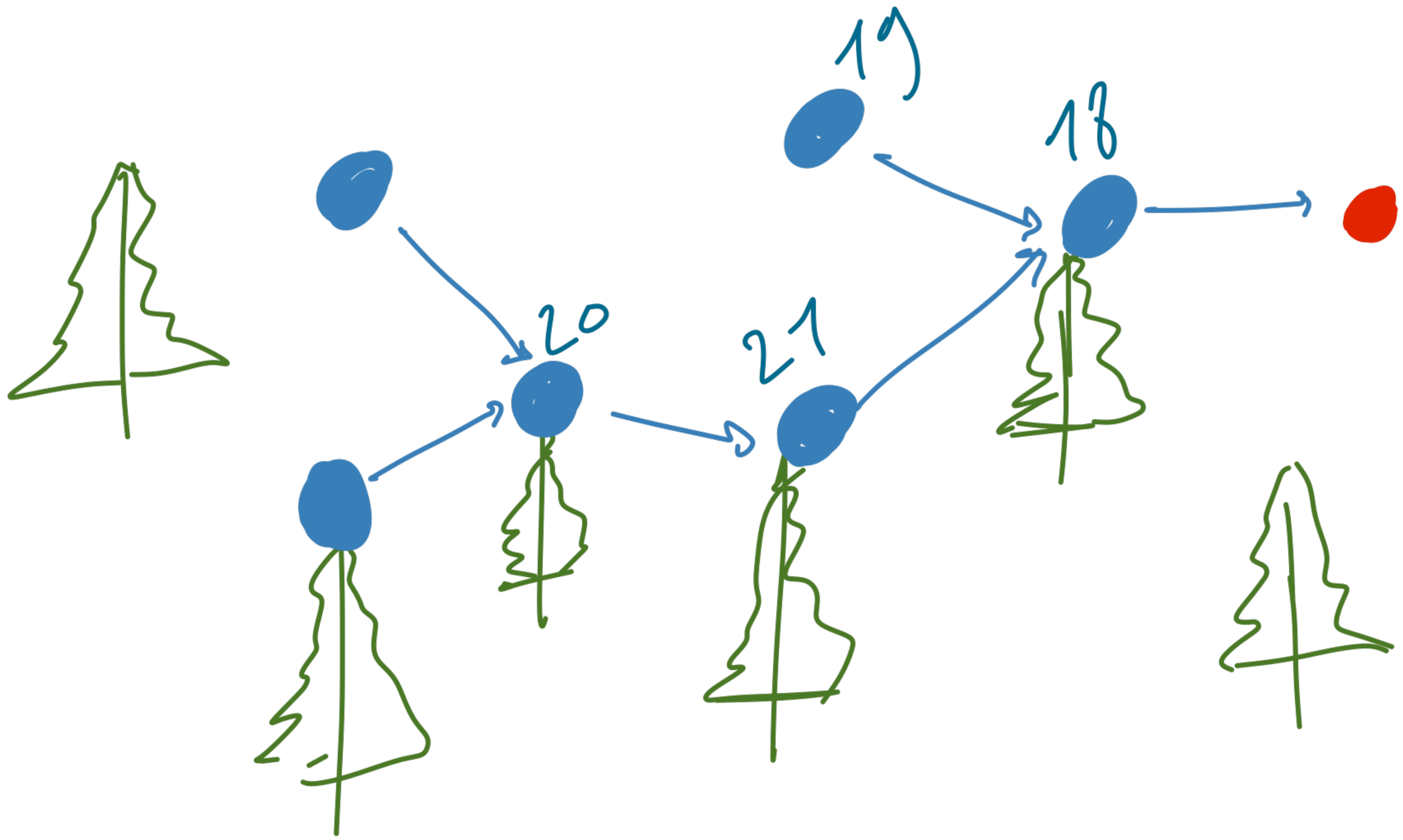
Data Aggregation Problem



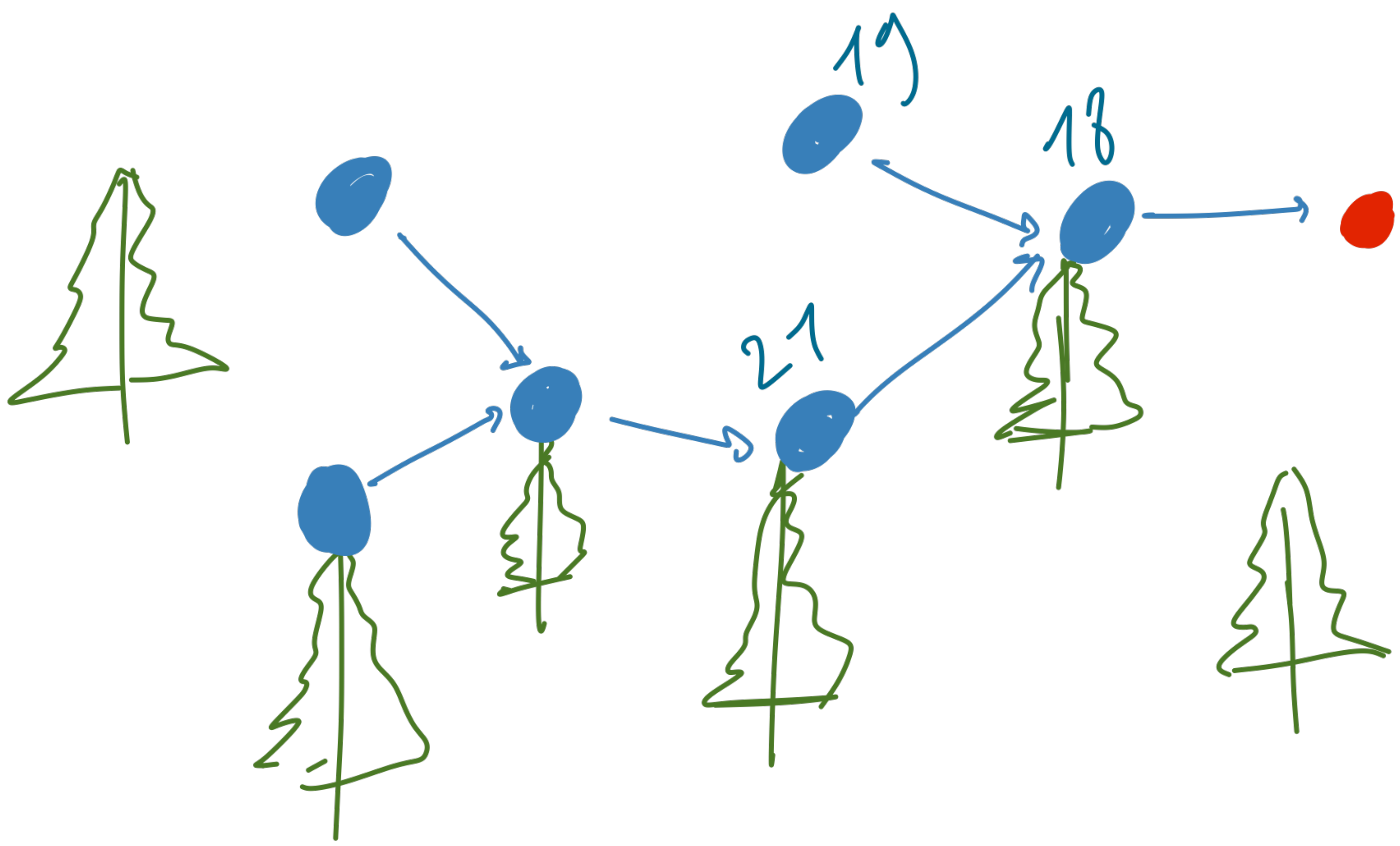
Data Aggregation Problem



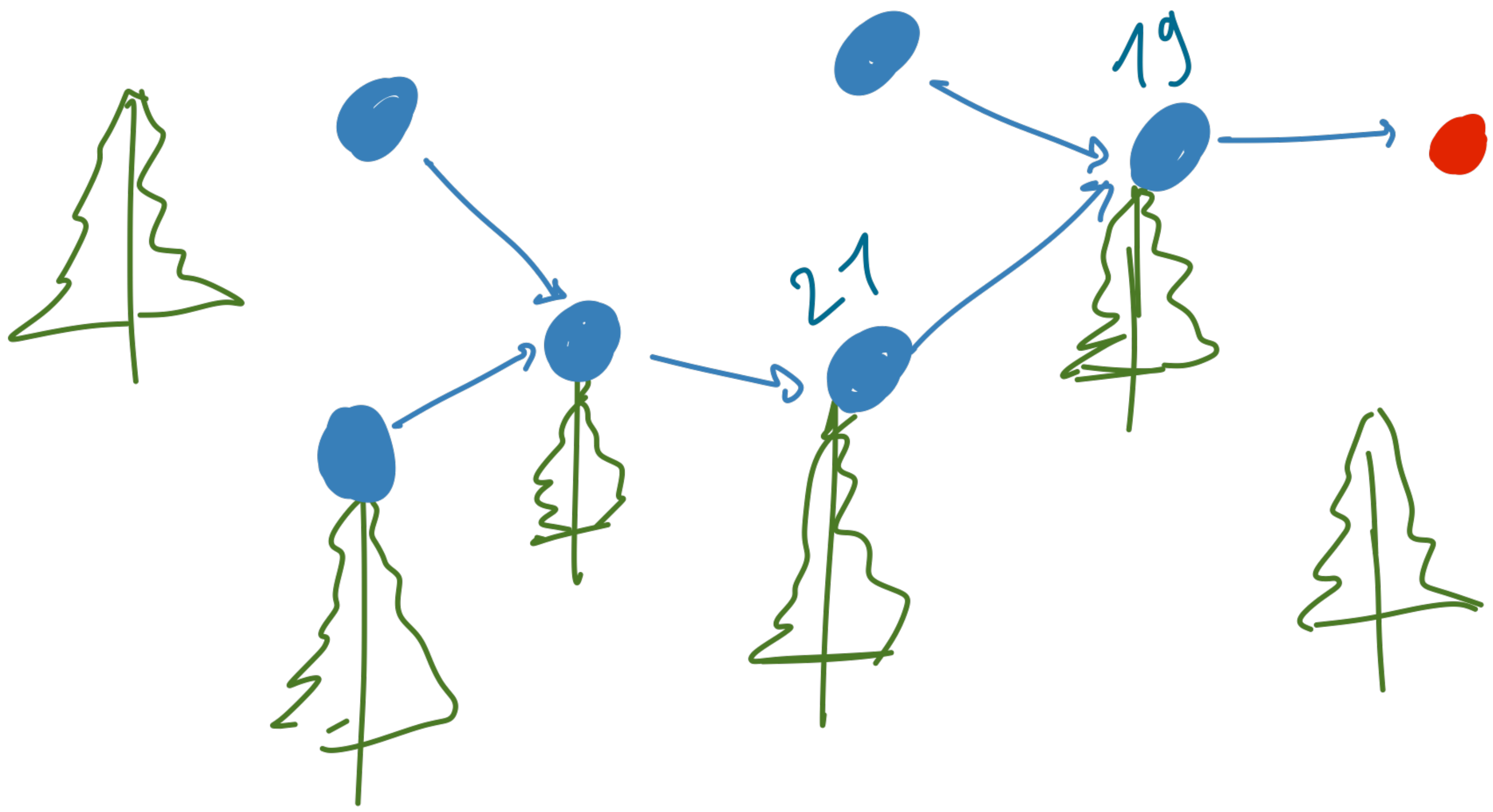
Data Aggregation Problem



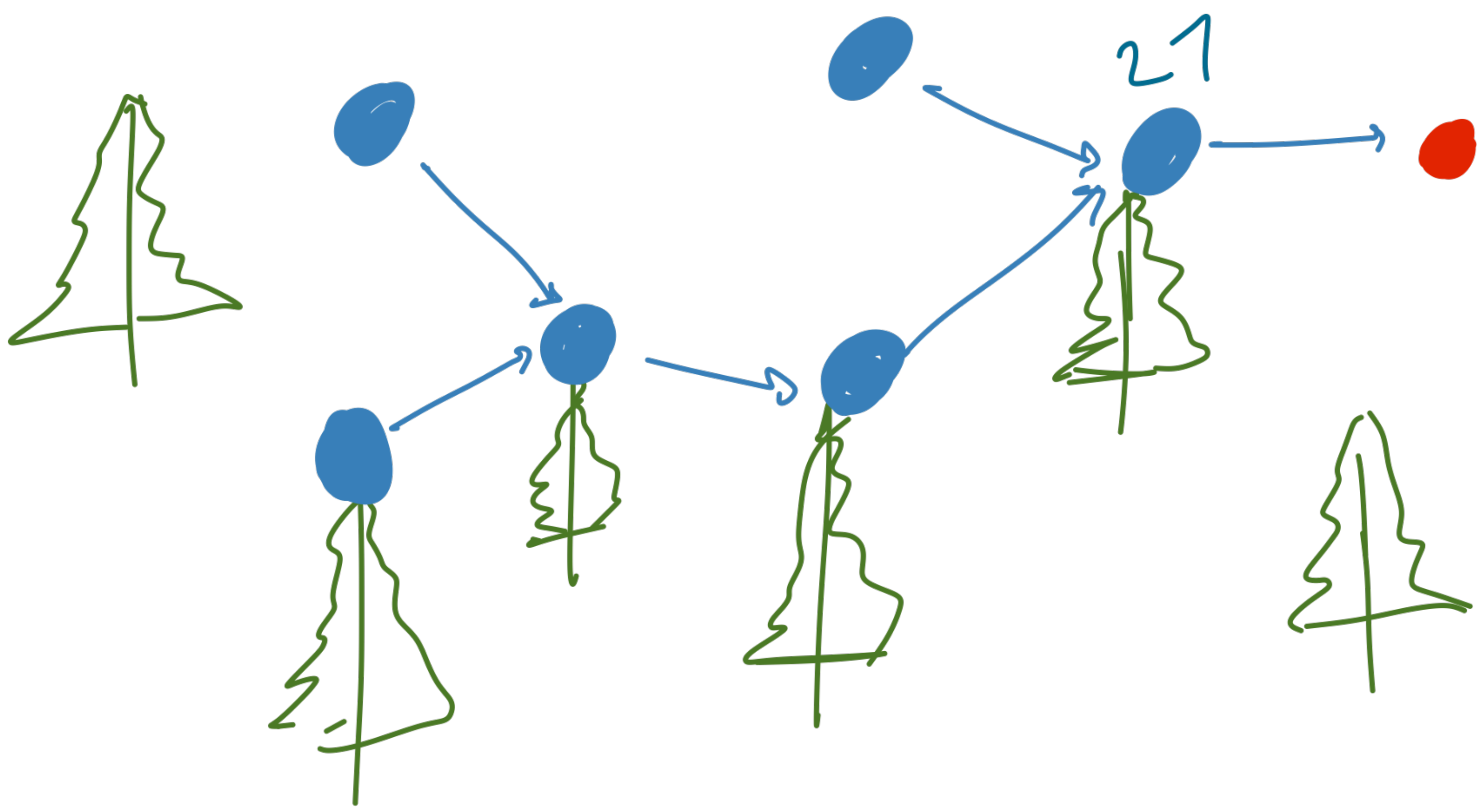
Data Aggregation Problem



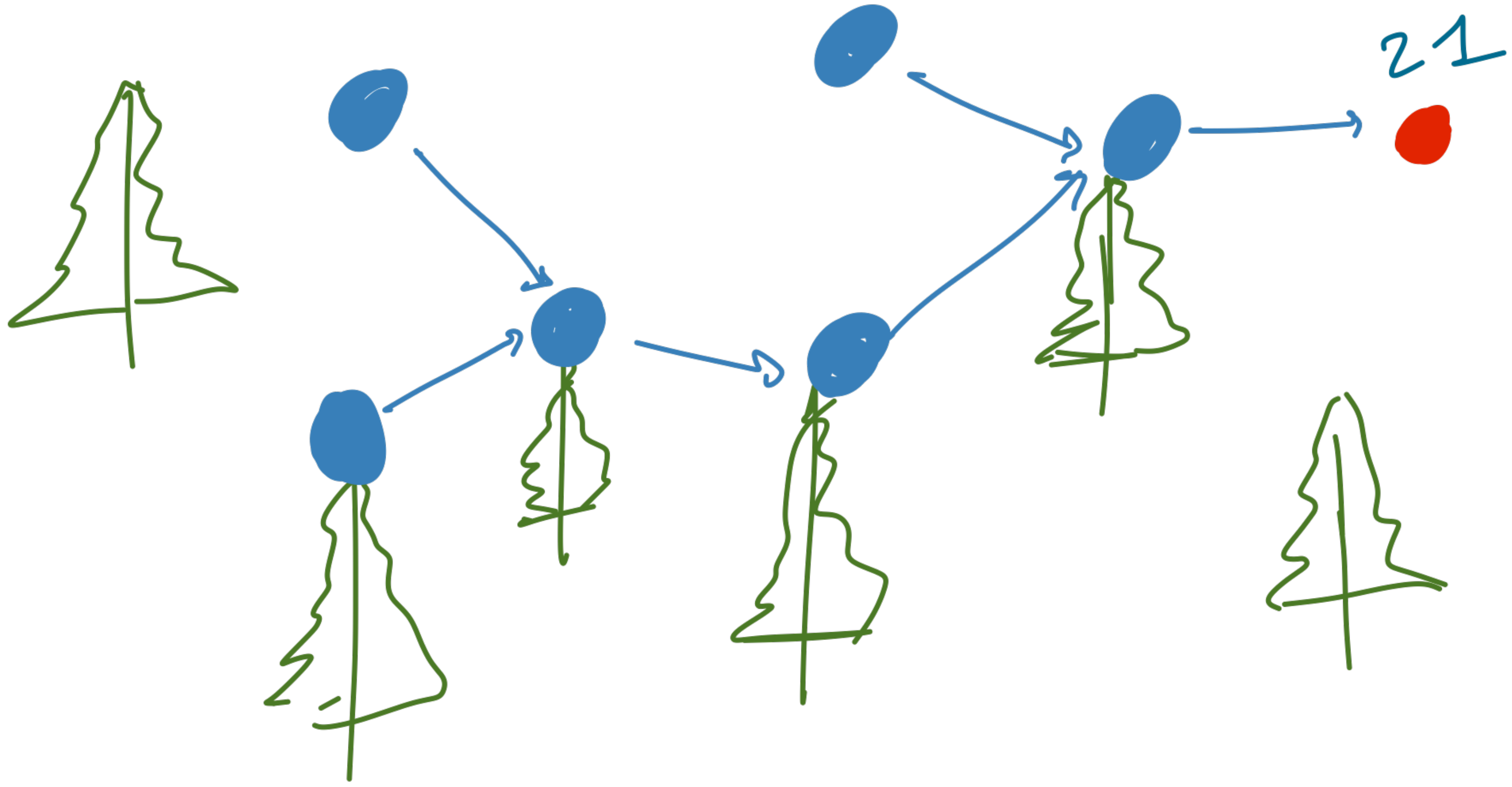
Data Aggregation Problem



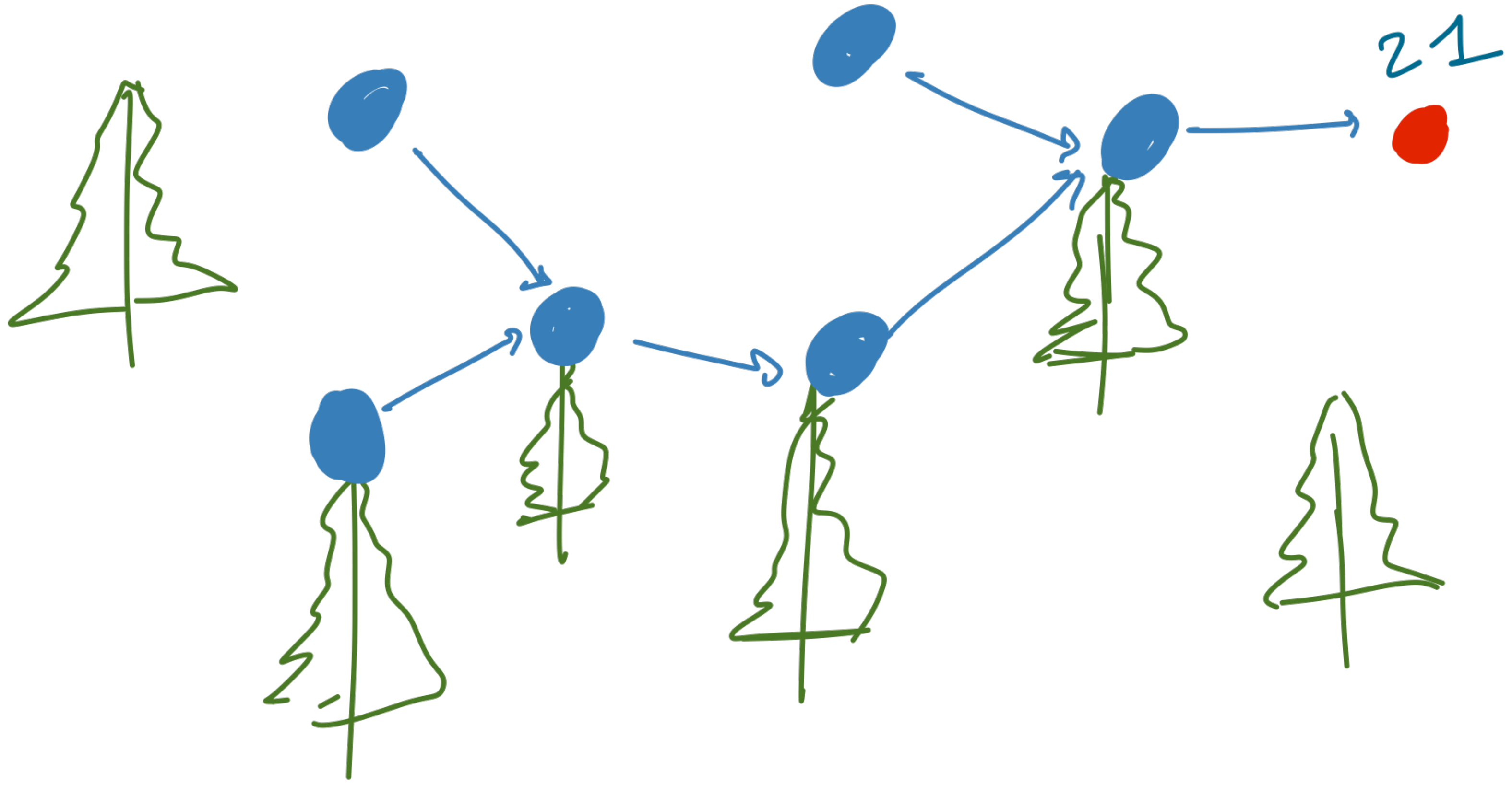
Data Aggregation Problem



Data Aggregation Problem



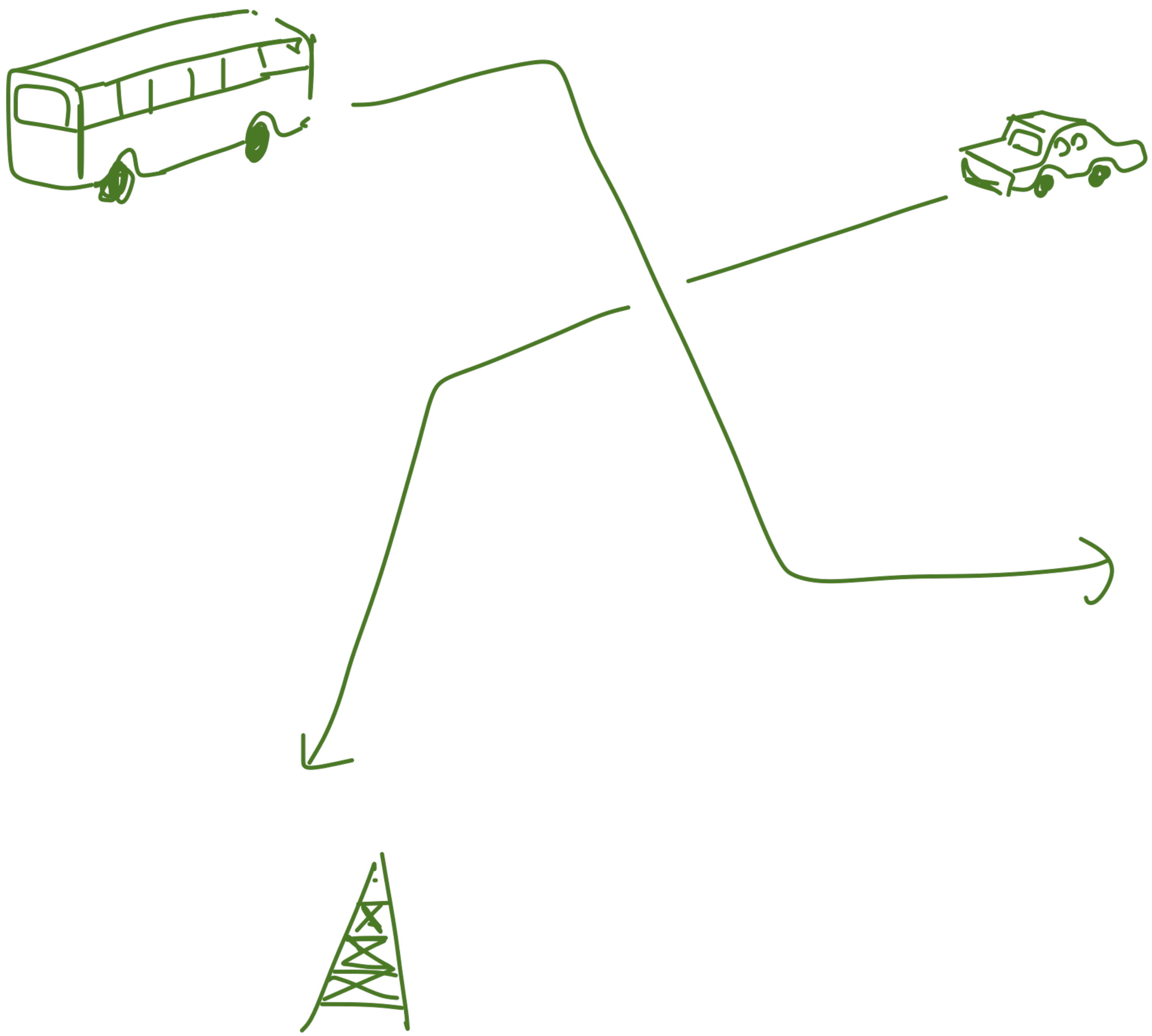
Data Aggregation Problem



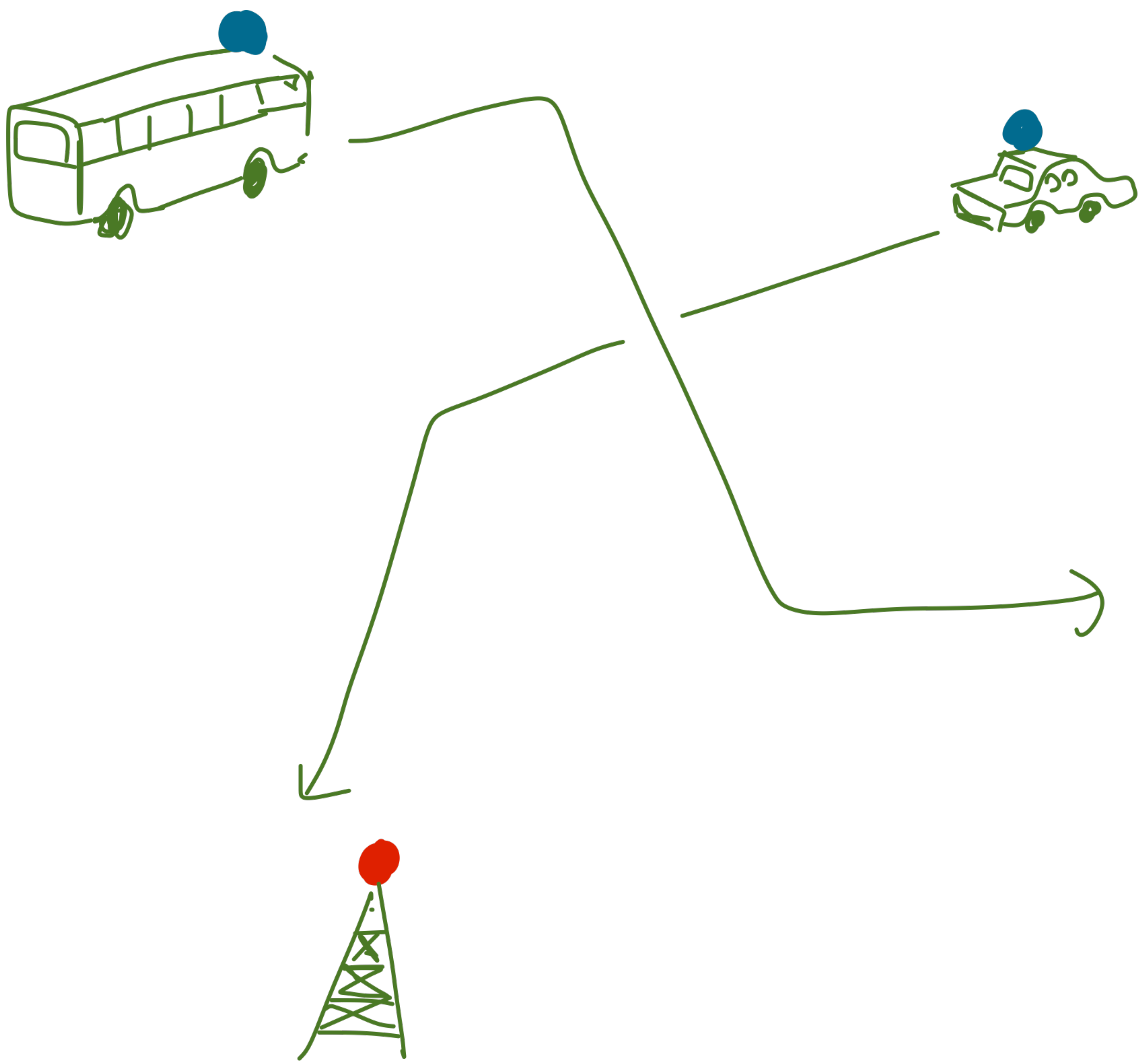
Each node transmits at most once.

The goal: to minimize the duration

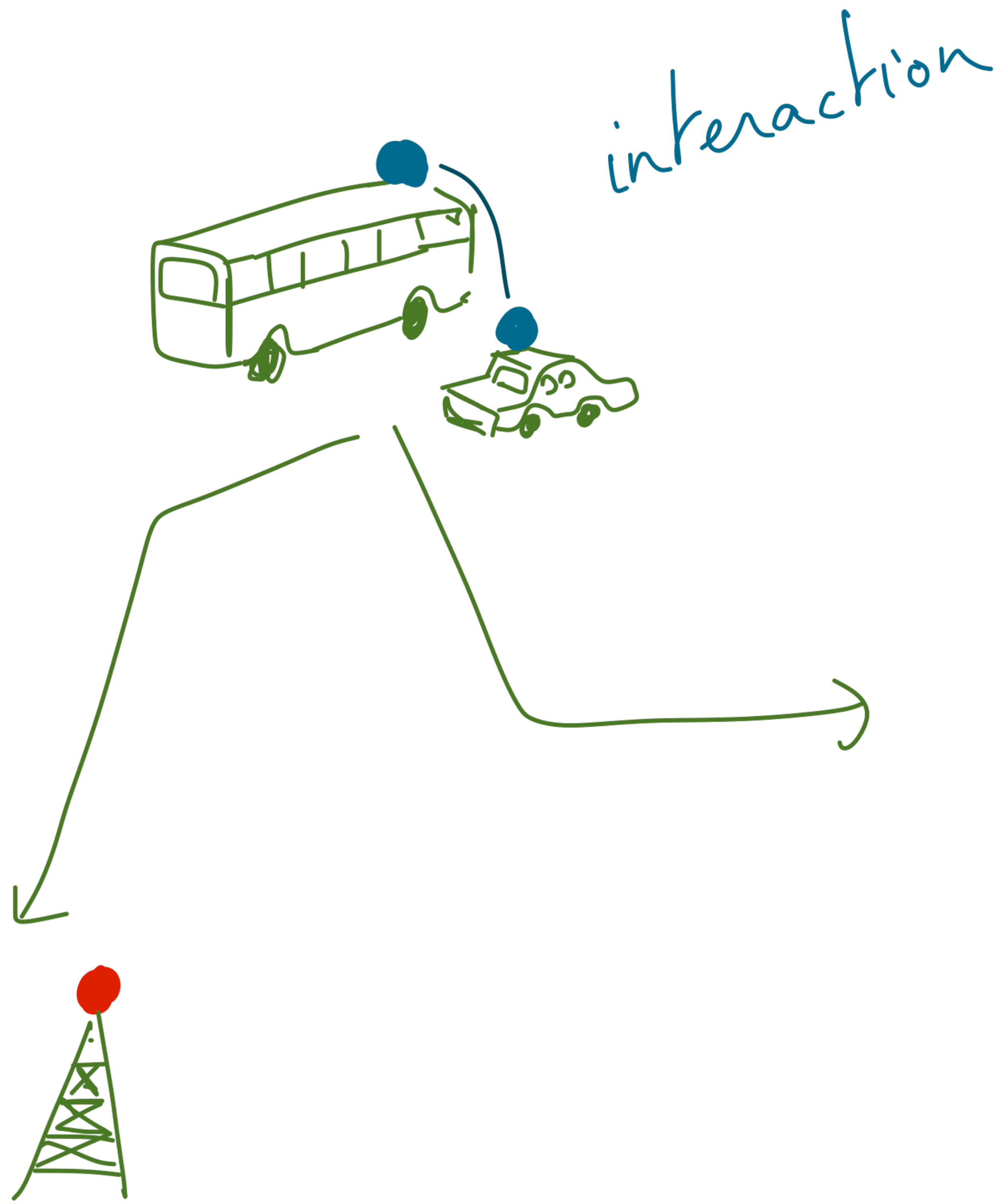
Data Aggregation Problem



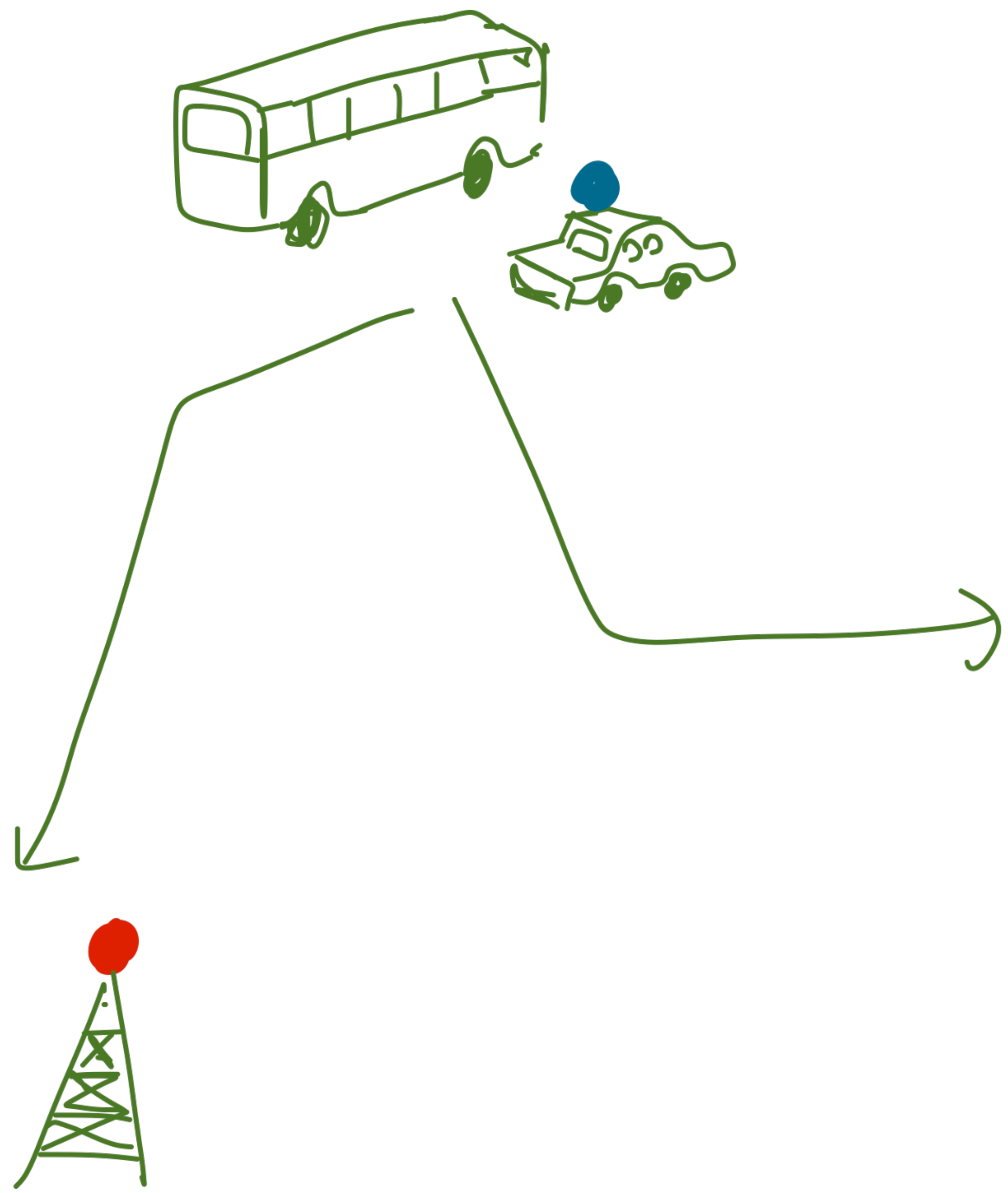
Data Aggregation Problem



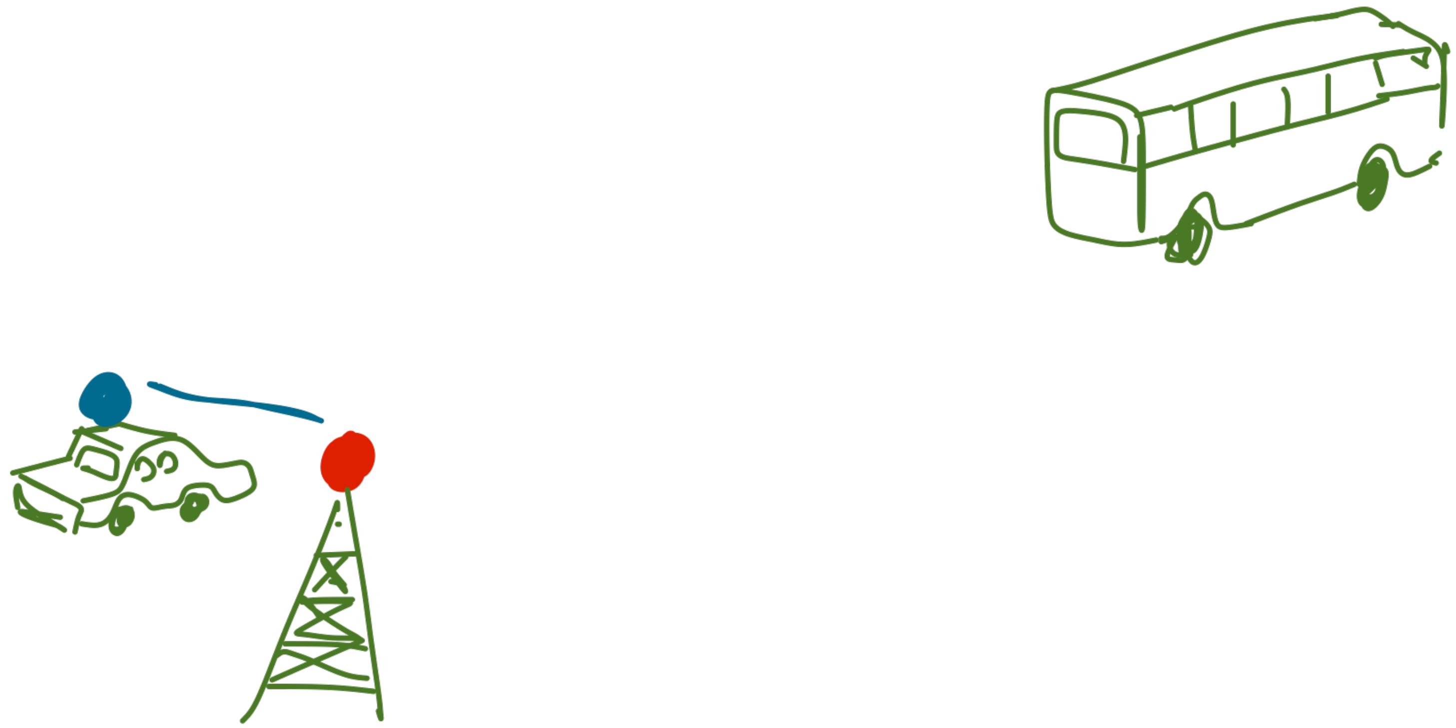
Data Aggregation Problem



Data Aggregation Problem



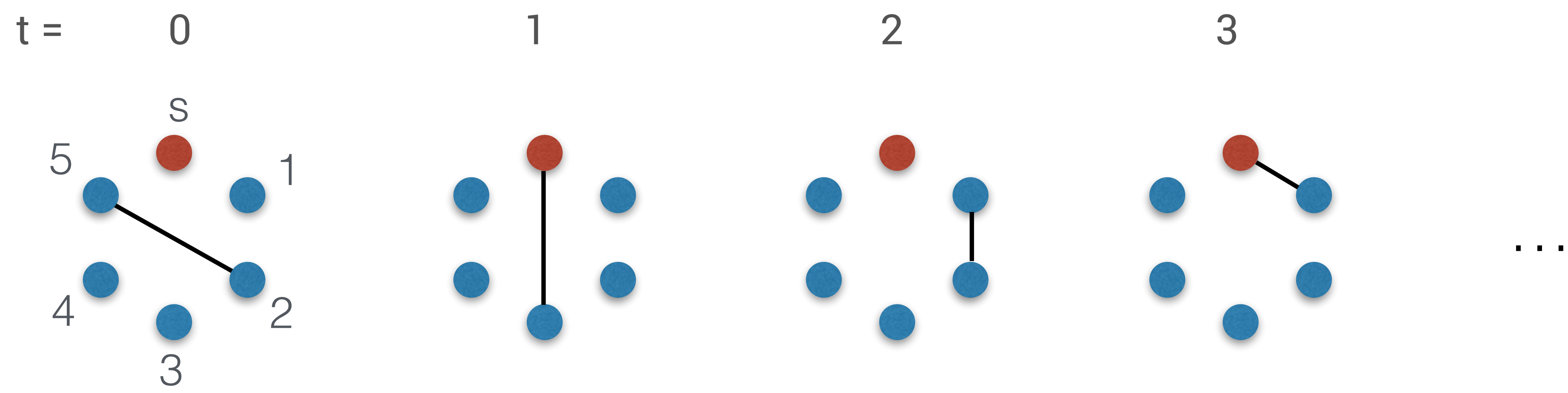
Data Aggregation Problem



Dynamic Data Aggregation Problem

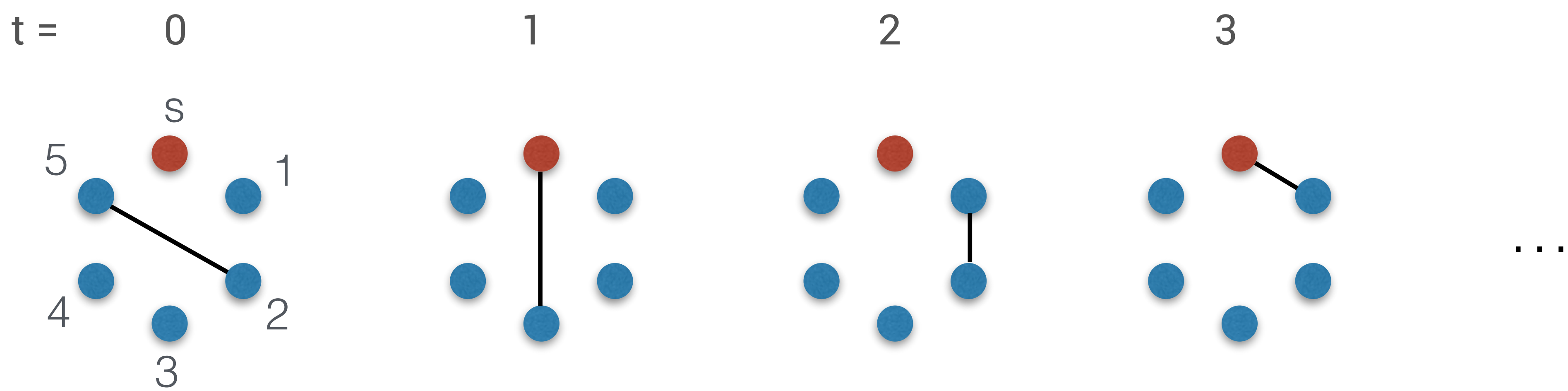
Dynamic Data Aggregation Problem

We consider a dynamic network with pairwise interactions



Dynamic Data Aggregation Problem

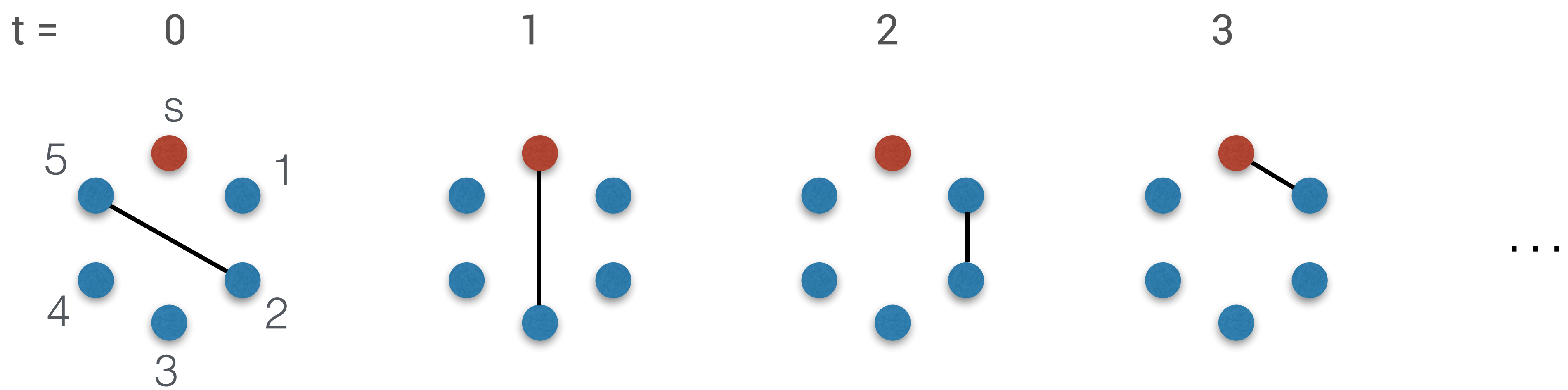
We consider a dynamic network with pairwise interactions



▷ We consider a sequence of interactions

Dynamic Data Aggregation Problem

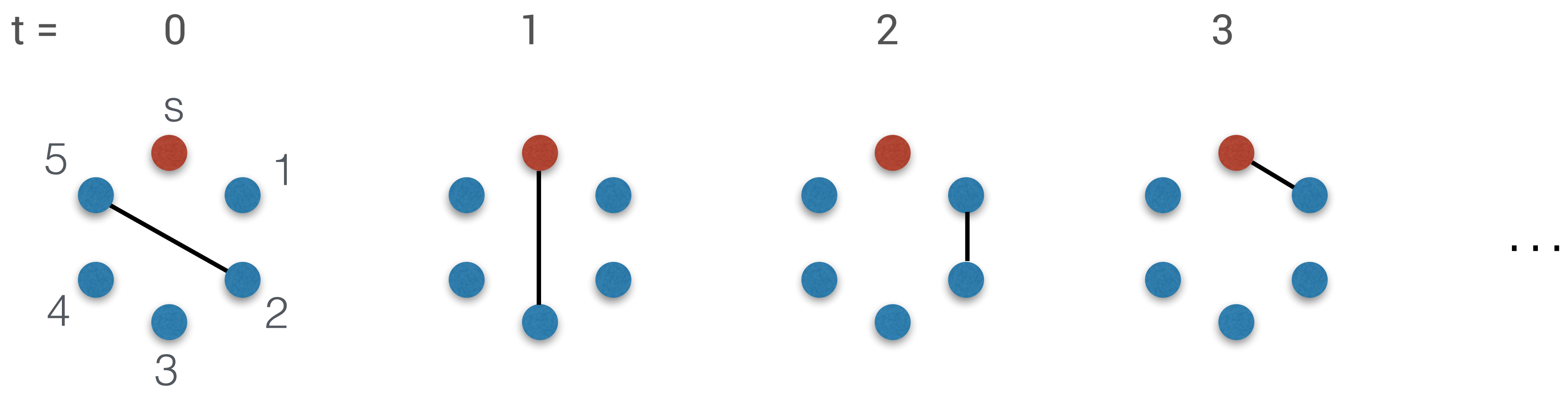
We consider a dynamic network with pairwise interactions



- We consider a sequence of interactions
- A node can transmit only once

Dynamic Data Aggregation Problem

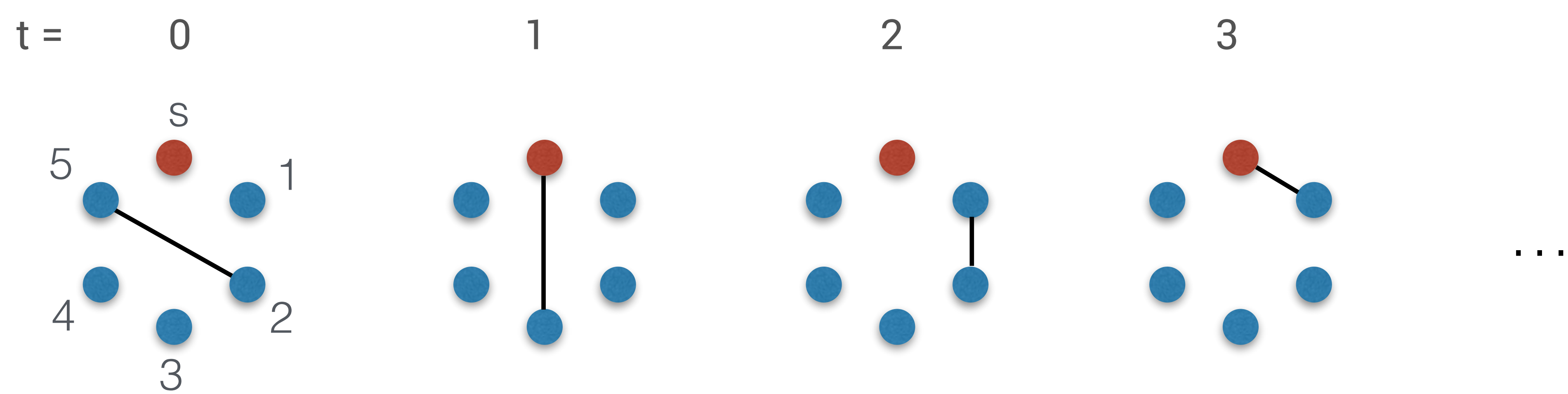
We consider a dynamic network with pairwise interactions



- We consider a sequence of interactions
- A node can transmit only once
- Nodes may or may not have other information

Dynamic Data Aggregation Problem

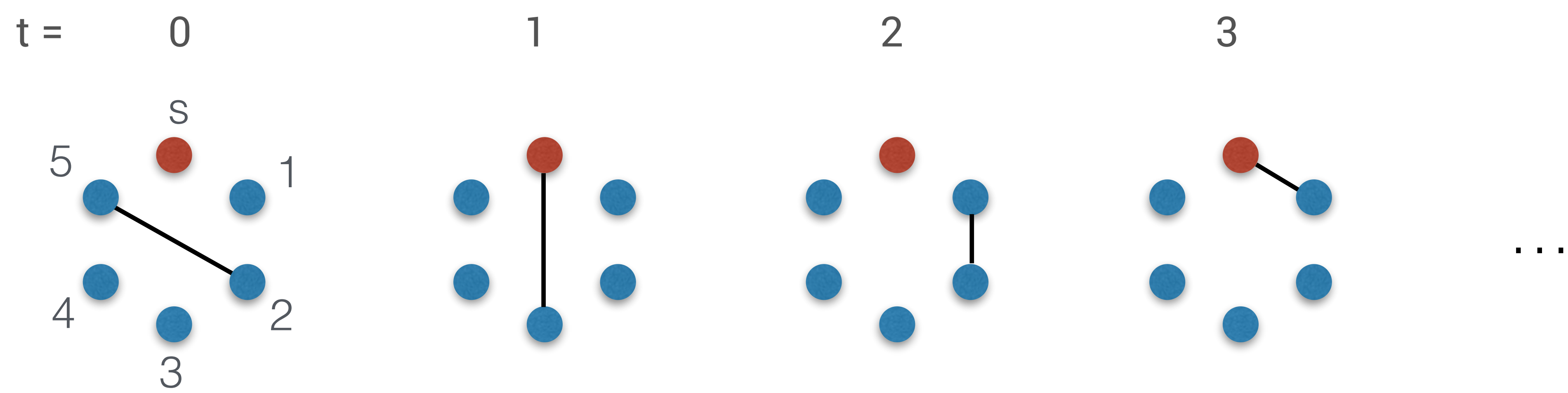
We consider a dynamic network with pairwise interactions



- We consider a sequence of interactions
- A node can transmit only once
- Nodes may or may not have other information
- The goal is to aggregate all the data with minimum duration

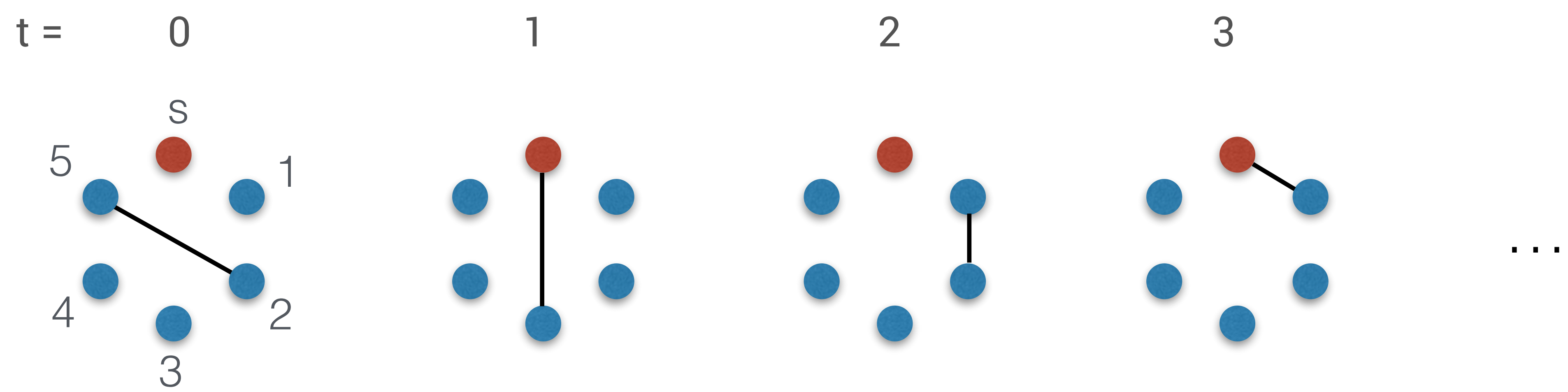
Distributed Online Data Aggregation

We consider a dynamic network with pairwise interactions



Distributed Online Data Aggregation

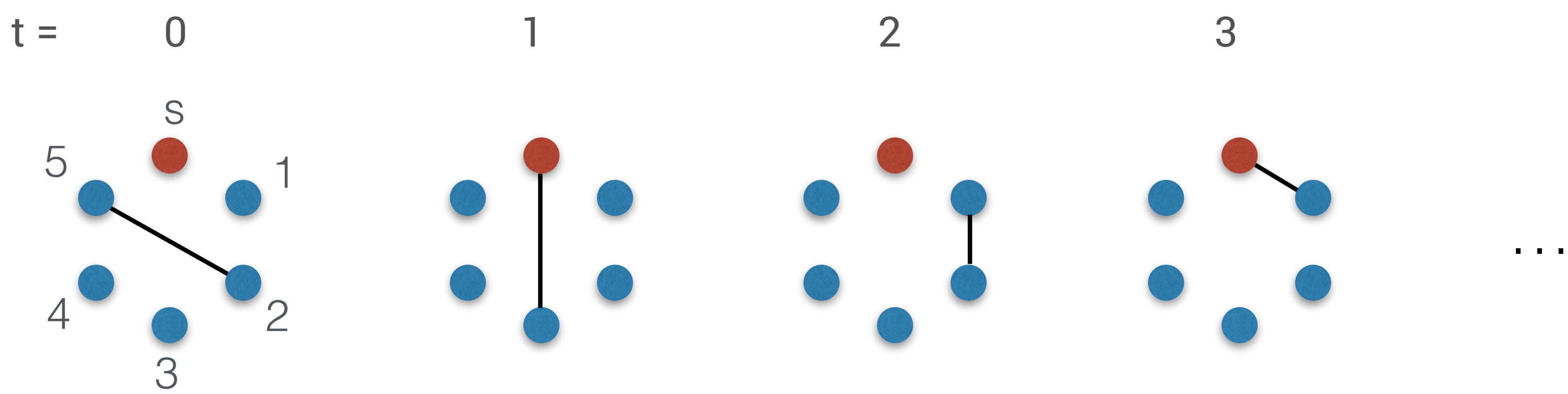
We consider a dynamic network with pairwise interactions



A Distributed Online Data Aggregation (DODA) Algorithm answers the question: Which node transmits?

Distributed Online Data Aggregation

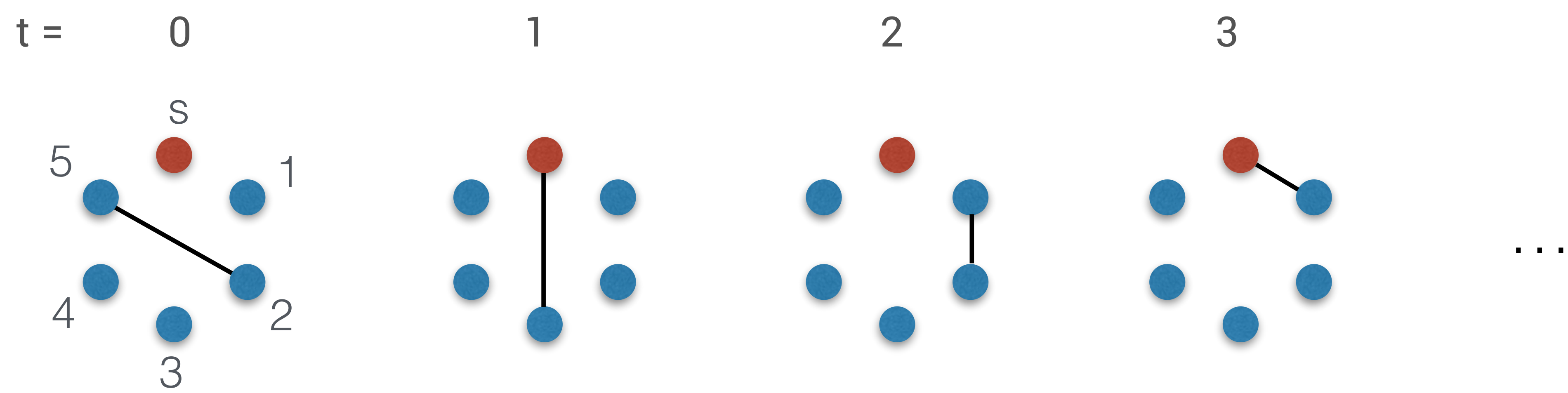
We consider a dynamic network with pairwise interactions



A Distributed Online Data Aggregation (DODA) Algorithm answers the question: Which node transmits?

Distributed Online Data Aggregation

We consider a dynamic network with pairwise interactions



A Distributed Online Data Aggregation (DODA) Algorithm answers the question: Which node transmits?

a transmits
or b transmits
or no one transmits

(a node can transmit only once)

Distributed Online Data Aggregation

has data	1 2 3 4
has transmitted	

Distributed Online Data Aggregation

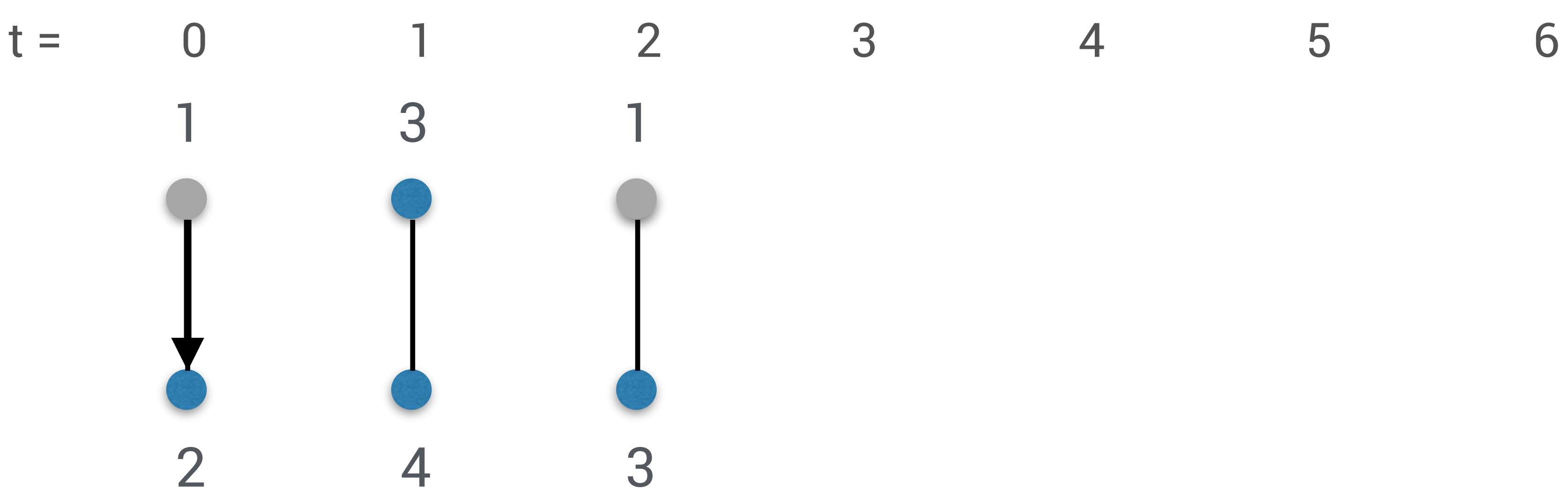
has data	2(1) 3 4
has transmitted	1

Distributed Online Data Aggregation

has data	2(1) 3 4
has transmitted	1

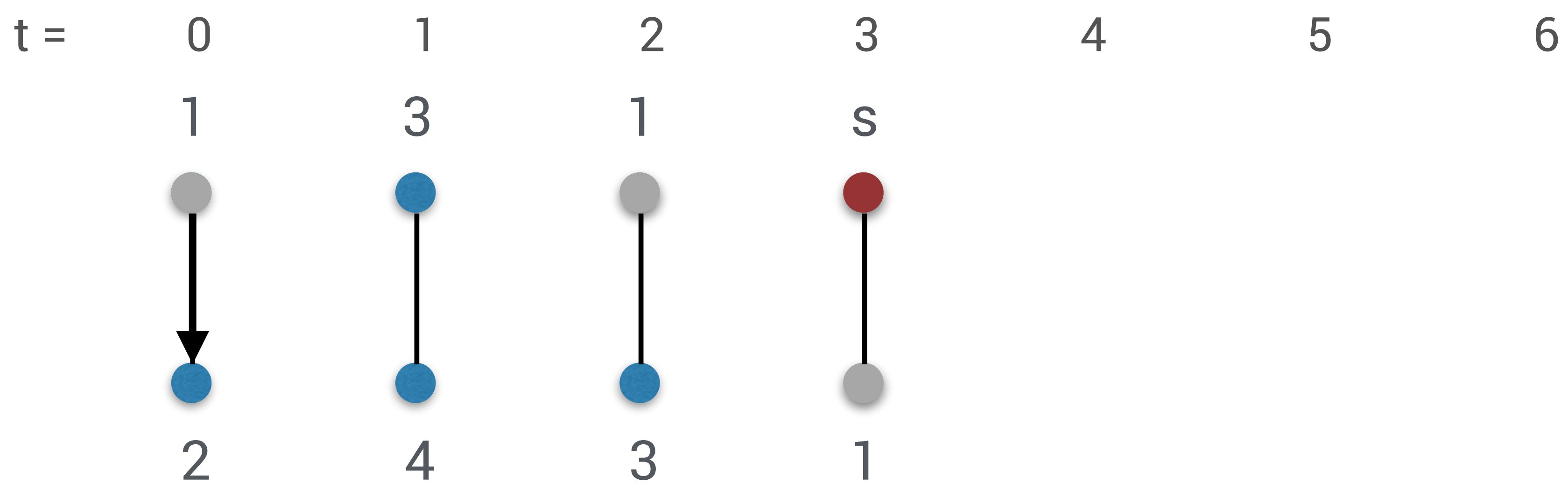
Distributed Online Data Aggregation

has data	2(1) 3 4
has transmitted	1



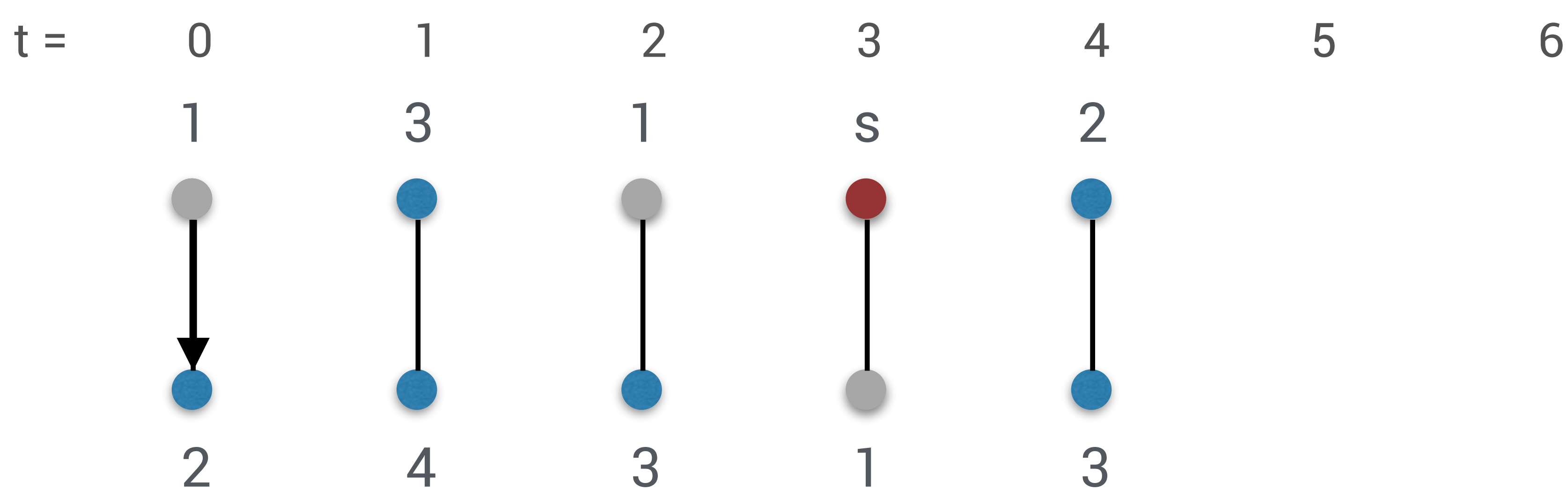
Distributed Online Data Aggregation

has data	2(1) 3 4
has transmitted	1



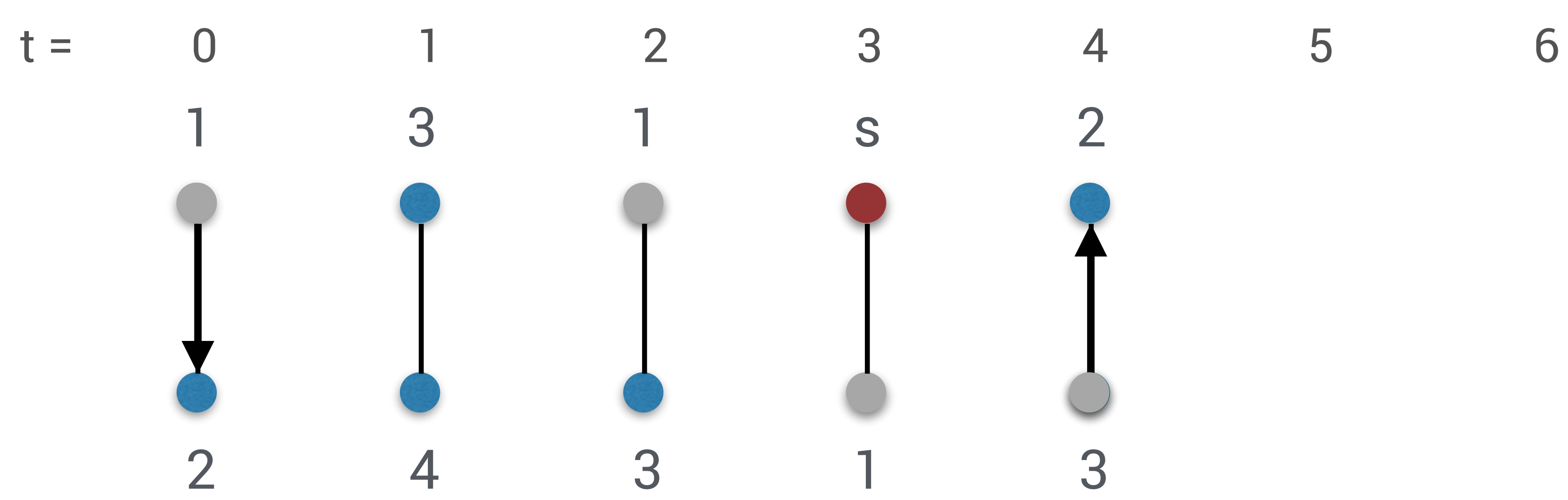
Distributed Online Data Aggregation

has data	2(1) 3 4
has transmitted	1



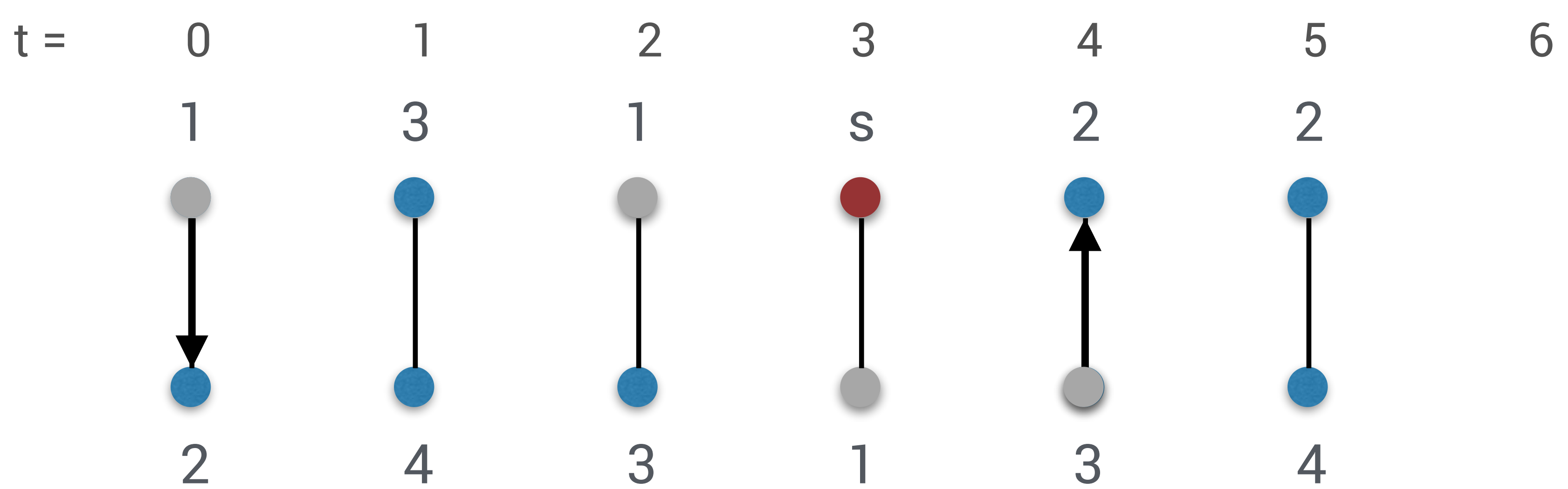
Distributed Online Data Aggregation

has data	2(1,3) 4
has transmitted	1 3



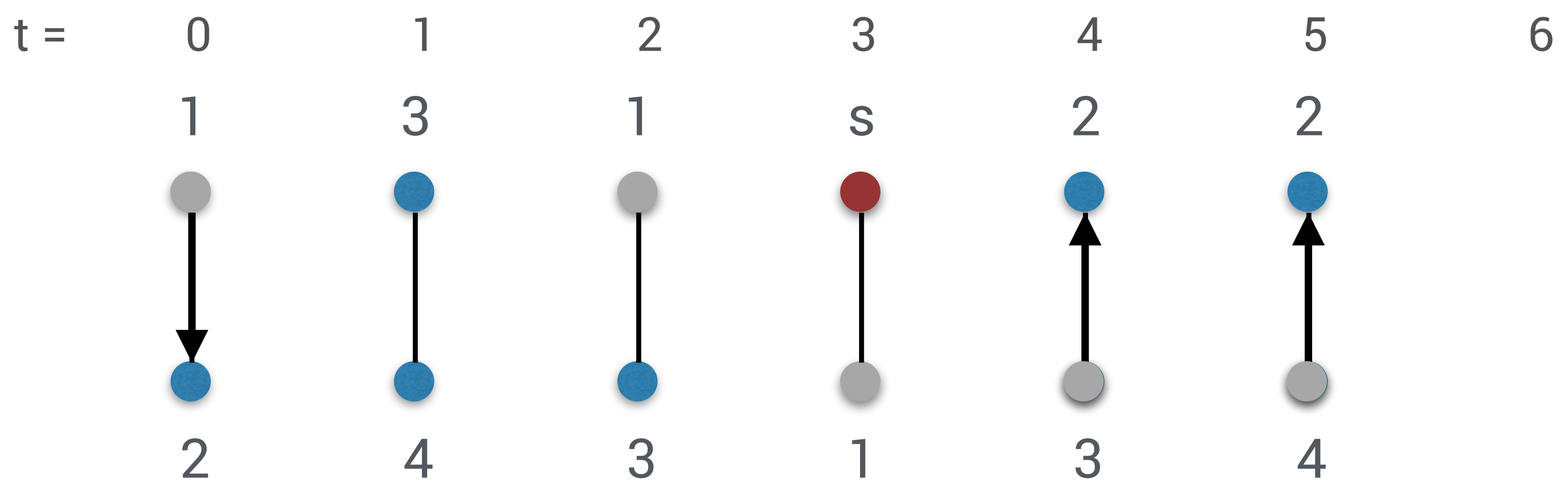
Distributed Online Data Aggregation

has data	2(1,3) 4
has transmitted	1 3



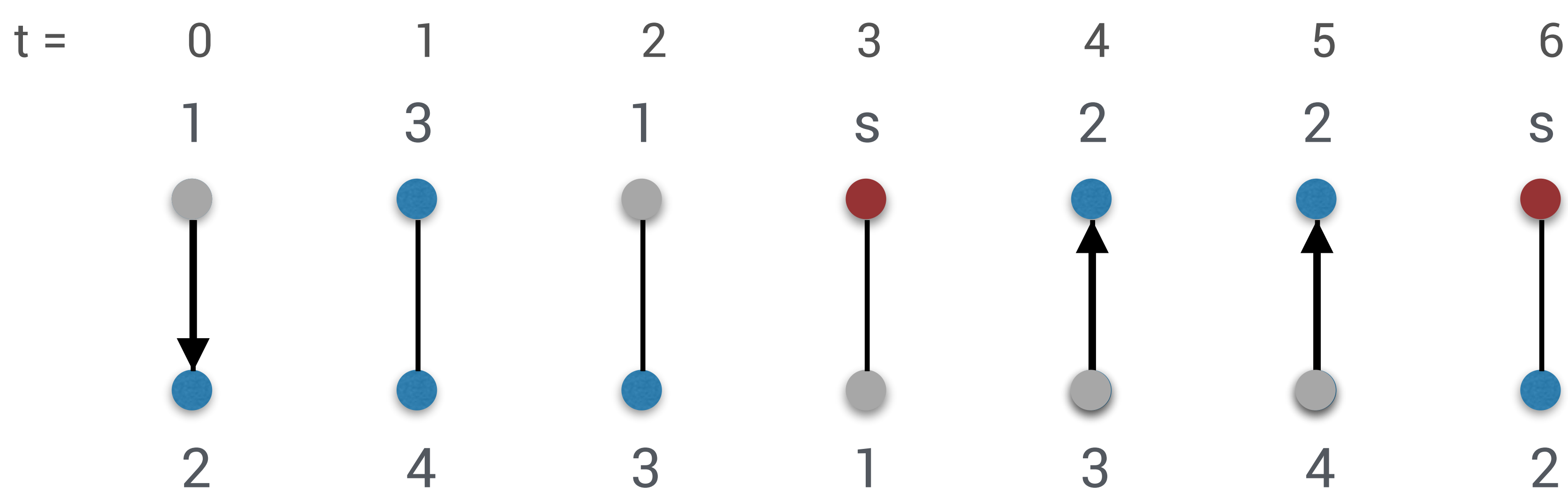
Distributed Online Data Aggregation

has data	2(1,3,4)
has transmitted	1 3 4



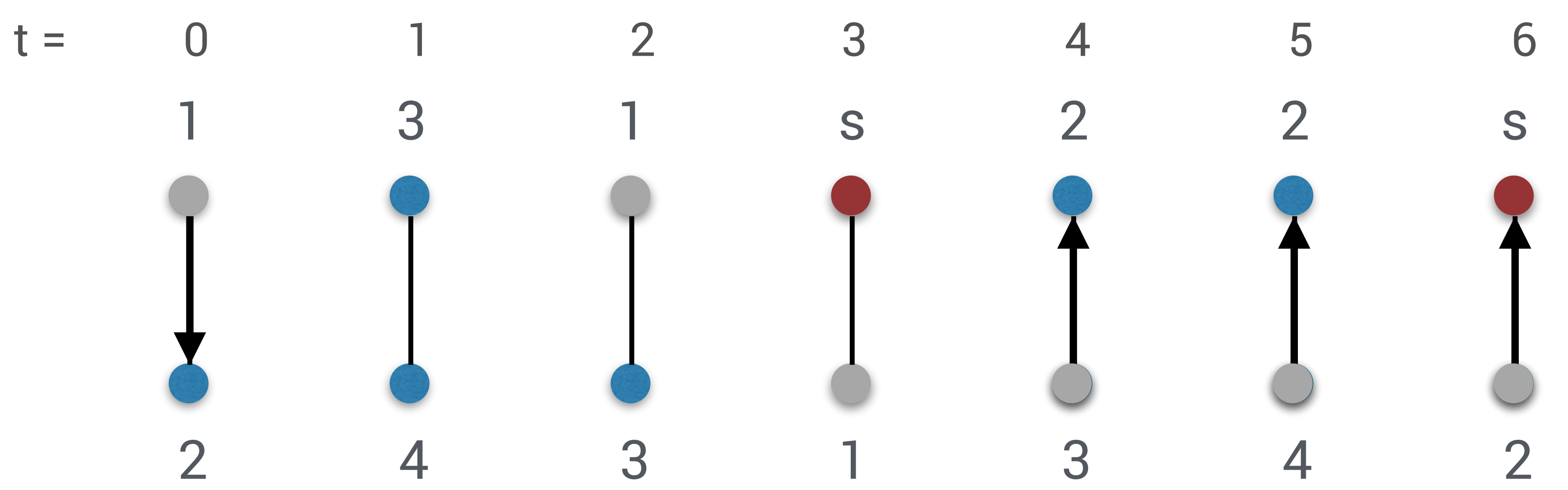
Distributed Online Data Aggregation

has data	2(1,3,4)
has transmitted	1 3 4



Distributed Online Data Aggregation

has data	
has transmitted	1 2 3 4



How to evaluate the performance of a DODA?

Is the duration of the aggregation is significant ?

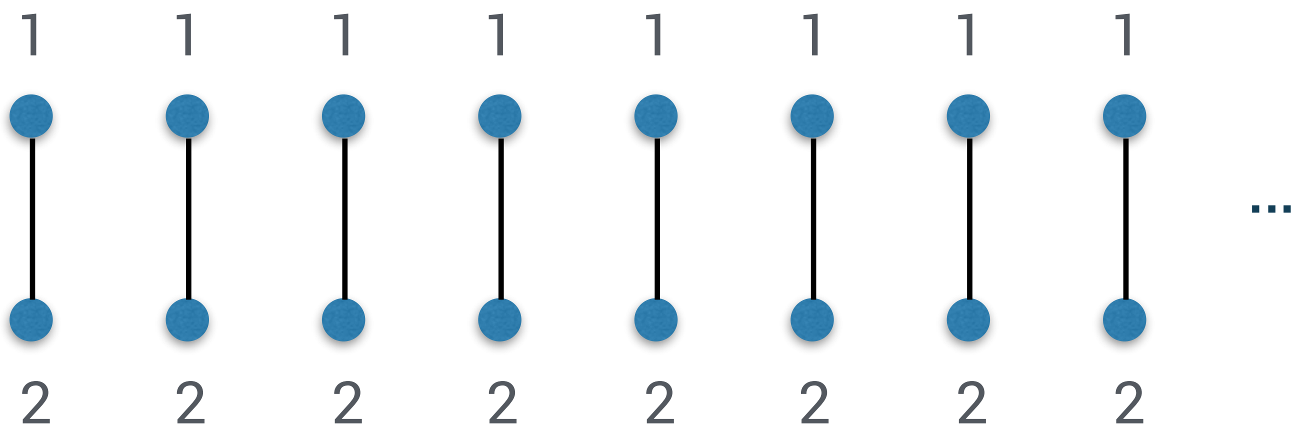
How to evaluate the performance of a DODA?

Is the duration of the aggregation is significant ? **No**

How to evaluate the performance of a DODA?

Is the duration of the aggregation is significant ? **No**

Even the offline optimal algorithm will struggle in the sequence:



How to evaluate the performance of a DODA?

Is the duration of the aggregation significant ? **No**

Is the ratio between the duration of the aggregation and the duration of the offline optimal algorithm significant ?

How to evaluate the performance of a DODA?

Is the duration of the aggregation significant ? **No**

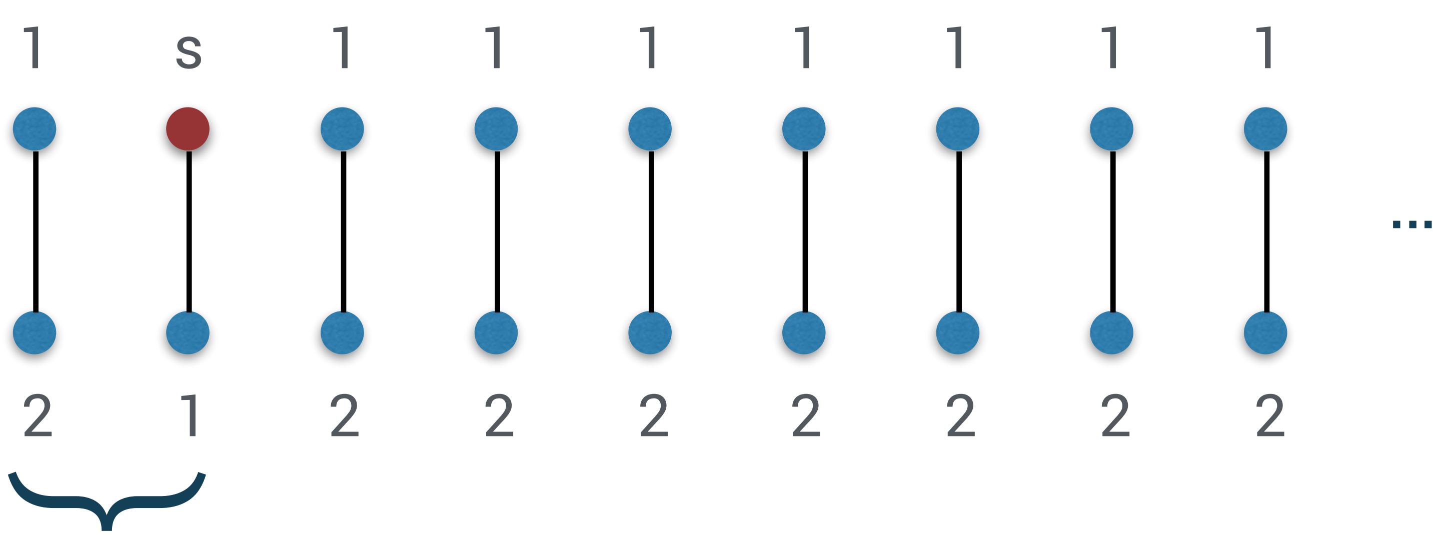
Is the ratio between the duration of the aggregation and the duration of the offline optimal algorithm significant ? **No**

How to evaluate the performance of a DODA?

Is the duration of the aggregation significant ? **No**

Is the ratio between the duration of the aggregation and the duration of the offline optimal algorithm significant ? **No**

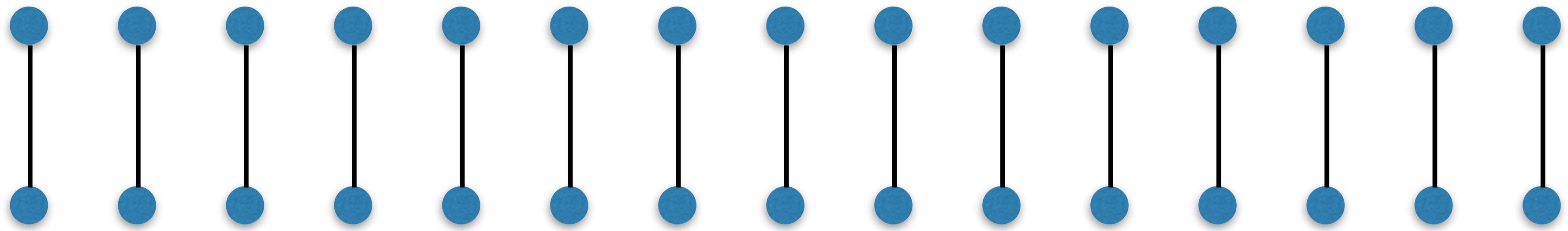
Each algorithm is either optimal or does not terminates in the sequence:



optimal duration = a convergecast

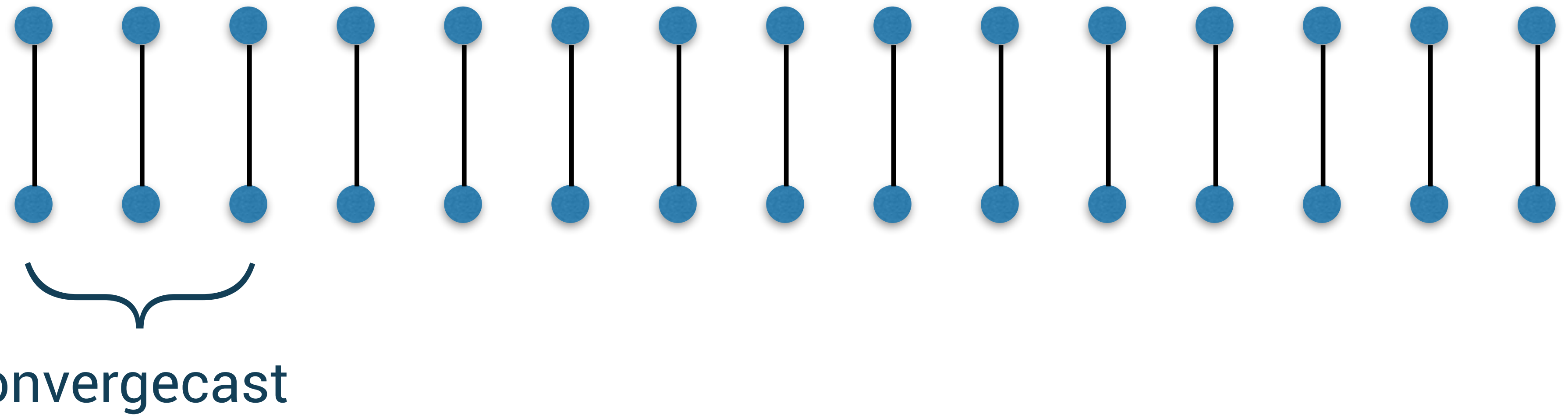
How to evaluate the performance of a DODA?

Definition of $cost_I(A)$, for an algorithm A in a sequence I :



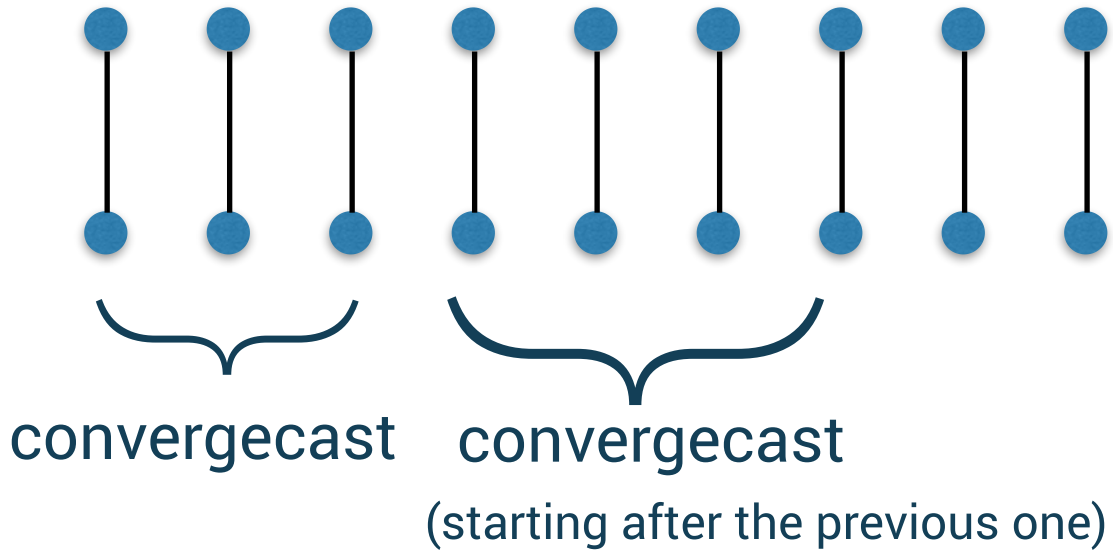
How to evaluate the performance of a DODA?

Definition of $cost_I(A)$, for an algorithm A in a sequence I :



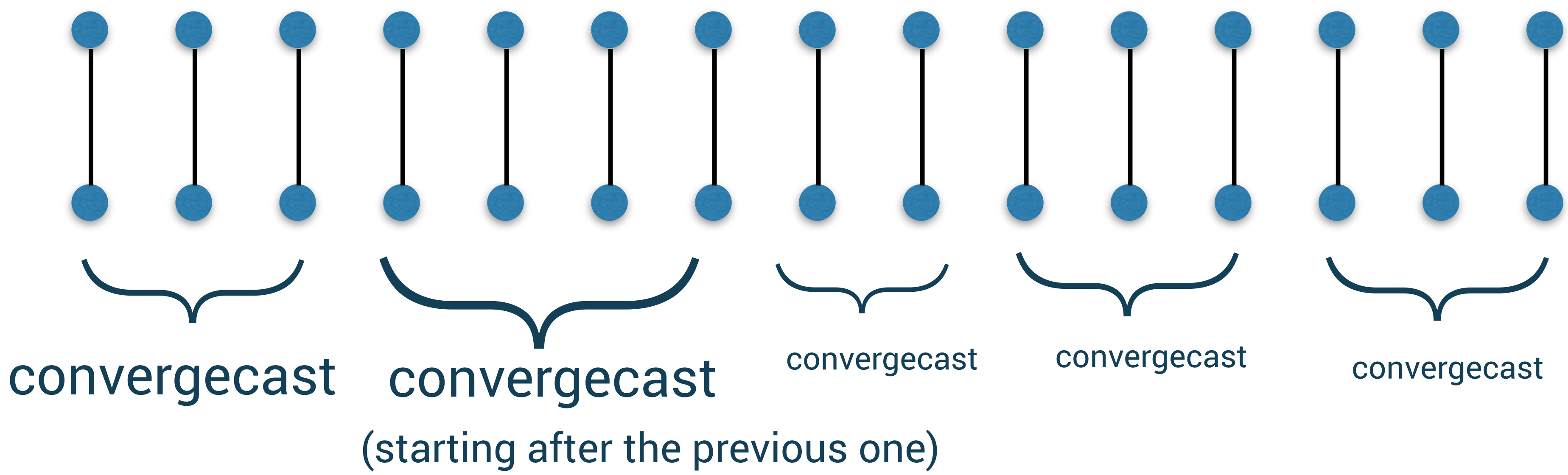
How to evaluate the performance of a DODA?

Definition of $cost_I(A)$, for an algorithm A in a sequence I :



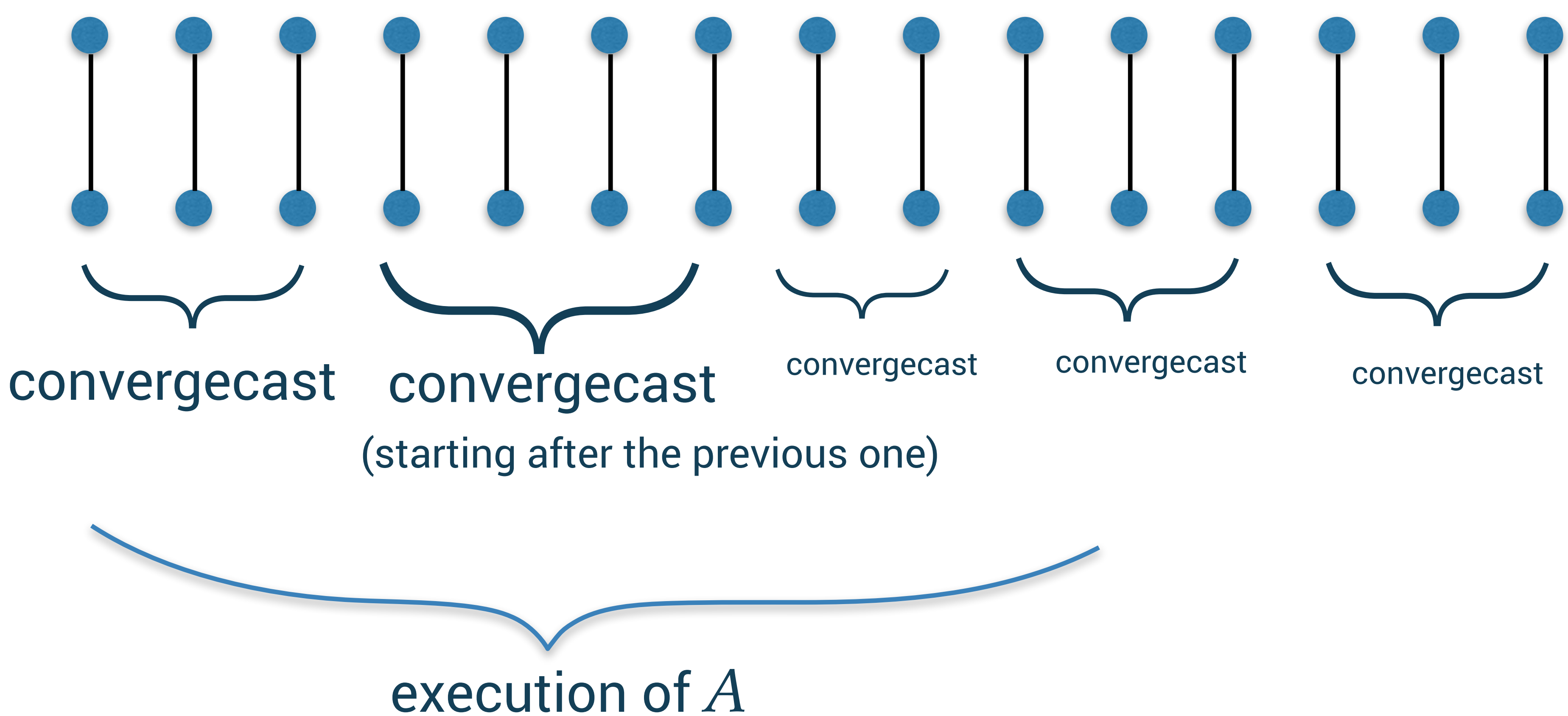
How to evaluate the performance of a DODA?

Definition of $cost_I(A)$, for an algorithm A in a sequence I :



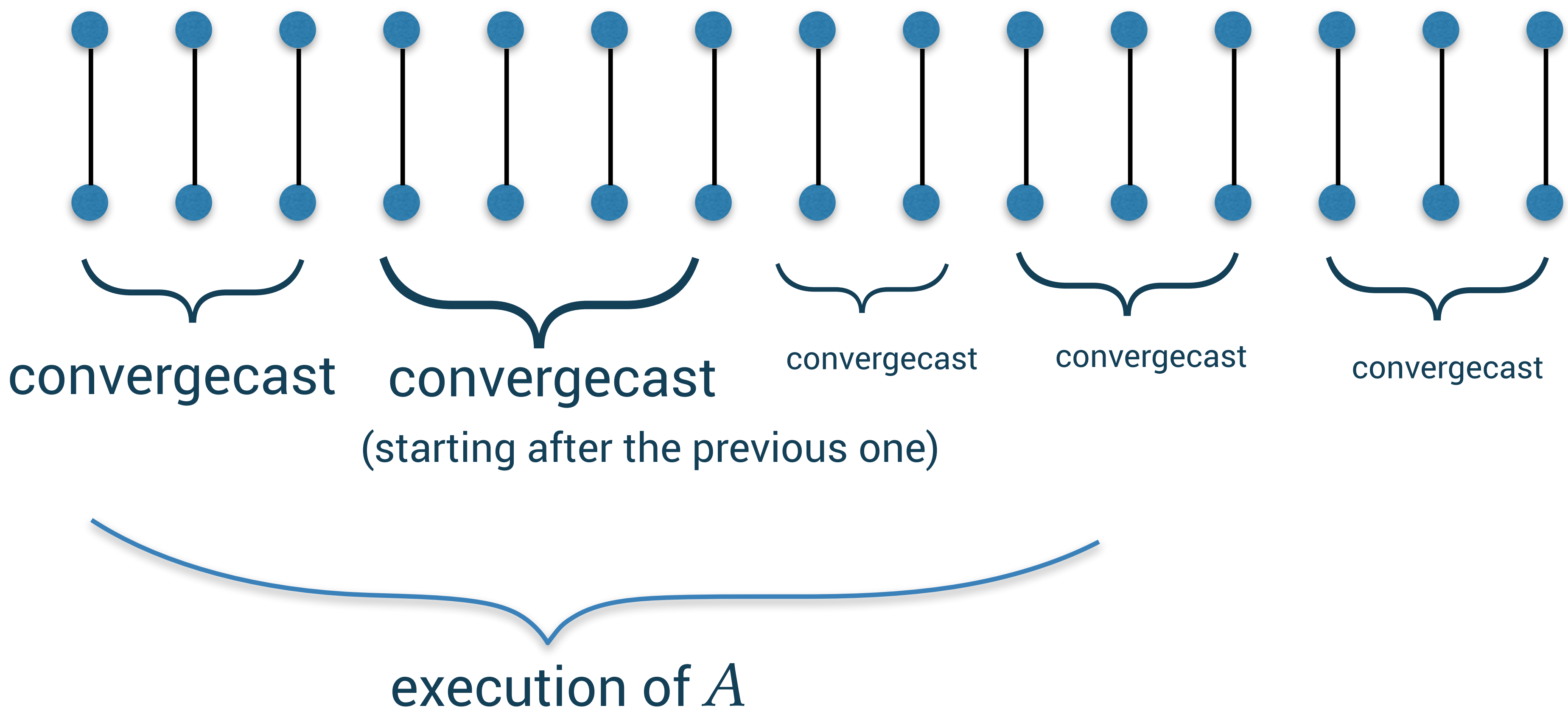
How to evaluate the performance of a DODA?

Definition of $cost_I(A)$, for an algorithm A in a sequence I :



How to evaluate the performance of a DODA?

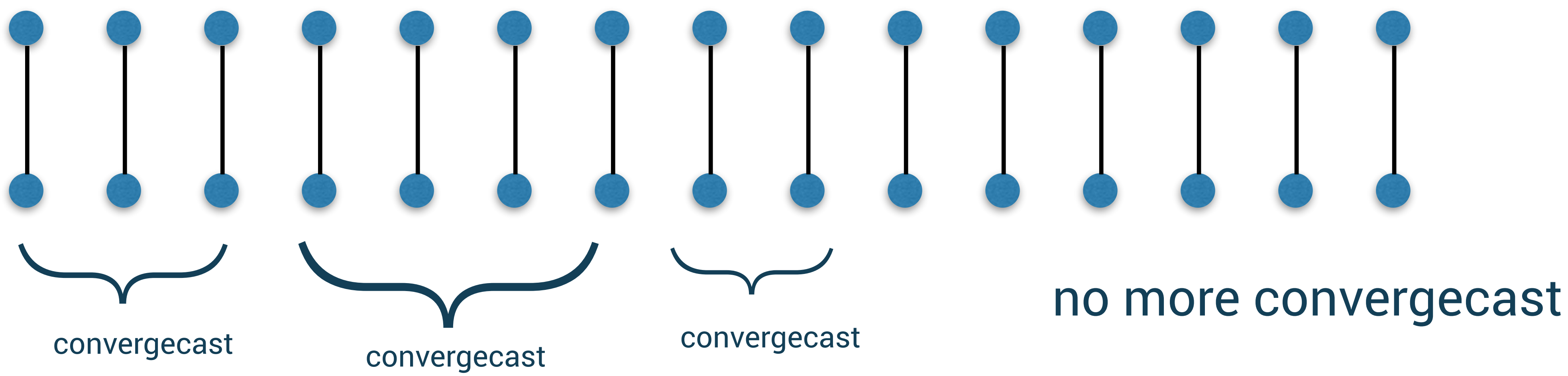
Definition of $cost_I(A)$, for an algorithm A in a sequence I :



$$cost_I(A) = 4$$

How to evaluate the performance of a DODA?

Definition of the $cost_I(A)$ function, for an algorithm A in a sequence I :

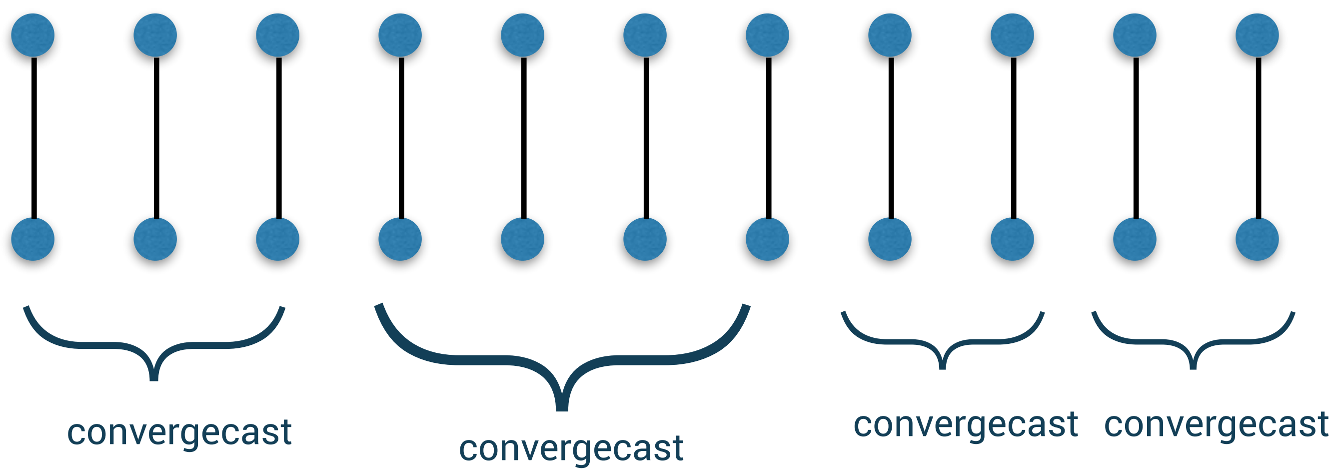


A does not terminate

$cost_I(A) = 4$

How to evaluate the performance of a DODA?

Definition of the $cost_I(A)$ function, for an algorithm A in a sequence I :

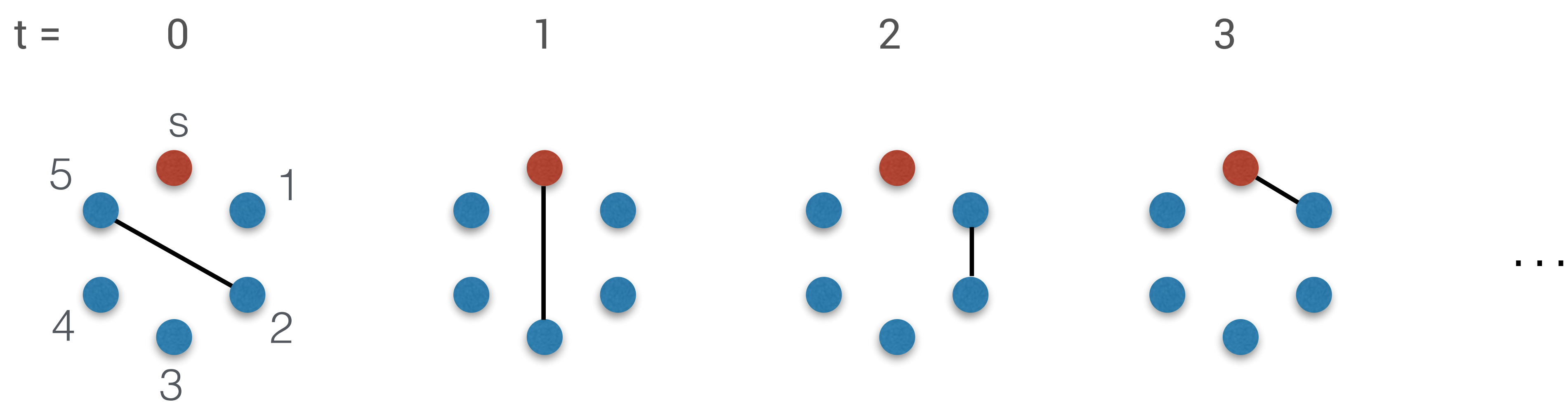


infinitely many convergecasts

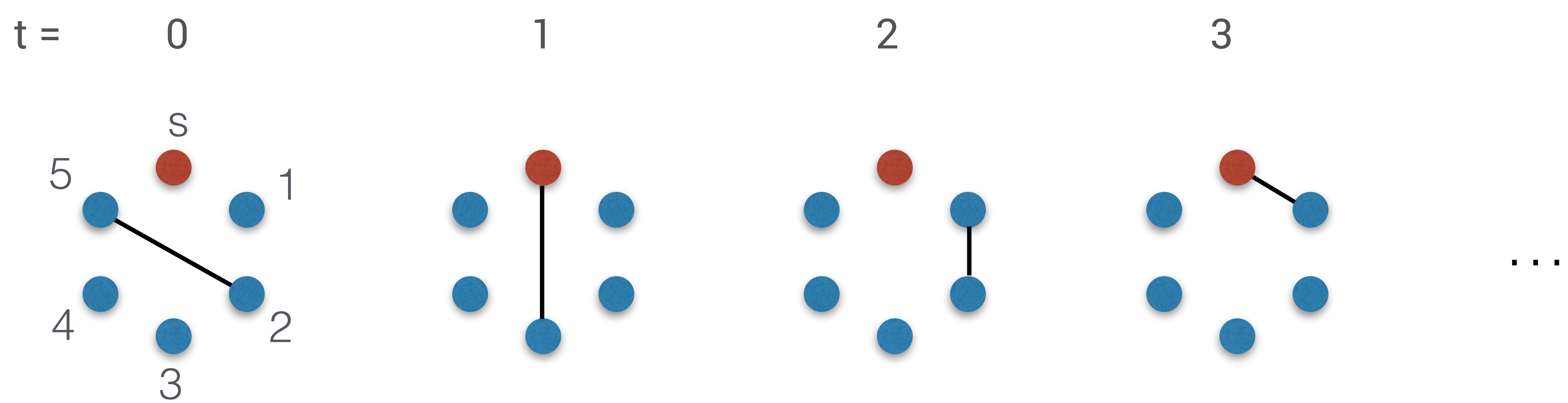
A does not terminate

$cost_I(A) = \infty$

Distributed Online Data Aggregation

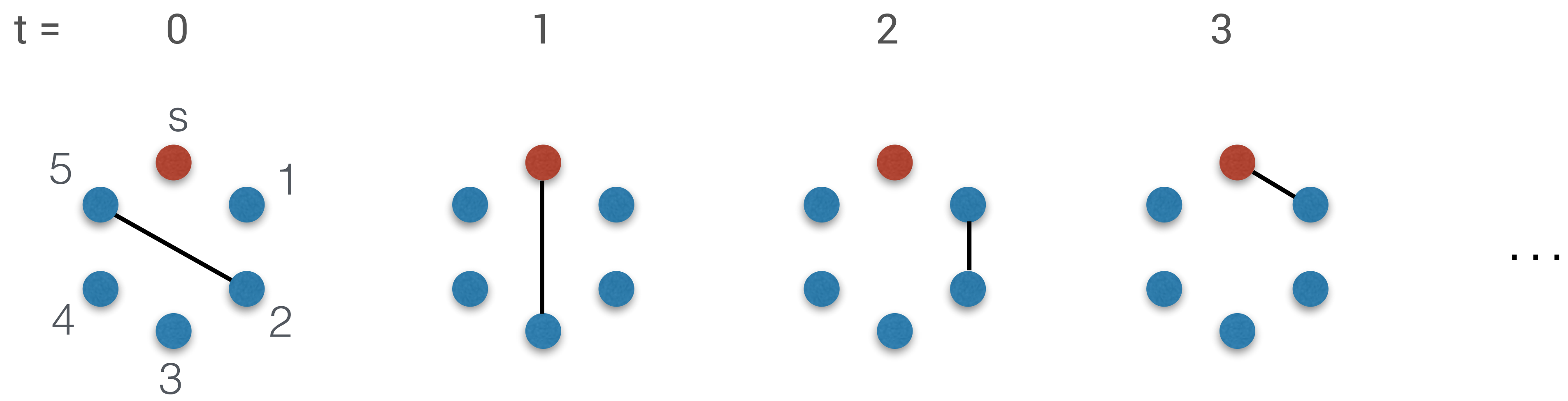


Distributed Online Data Aggregation



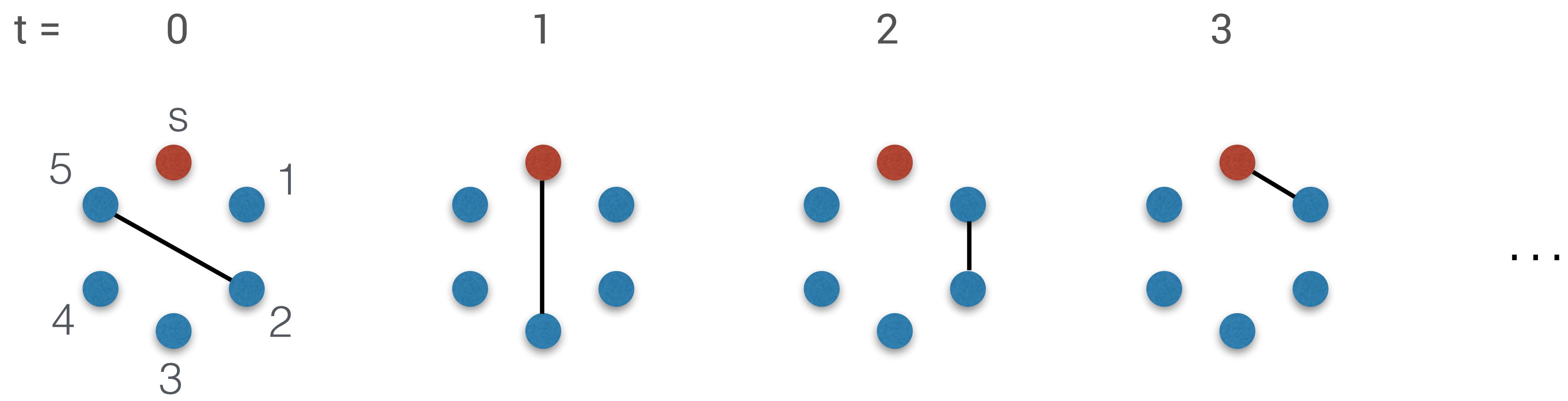
Who generates the sequence?

Distributed Online Data Aggregation



Who generates the sequence? An adversary

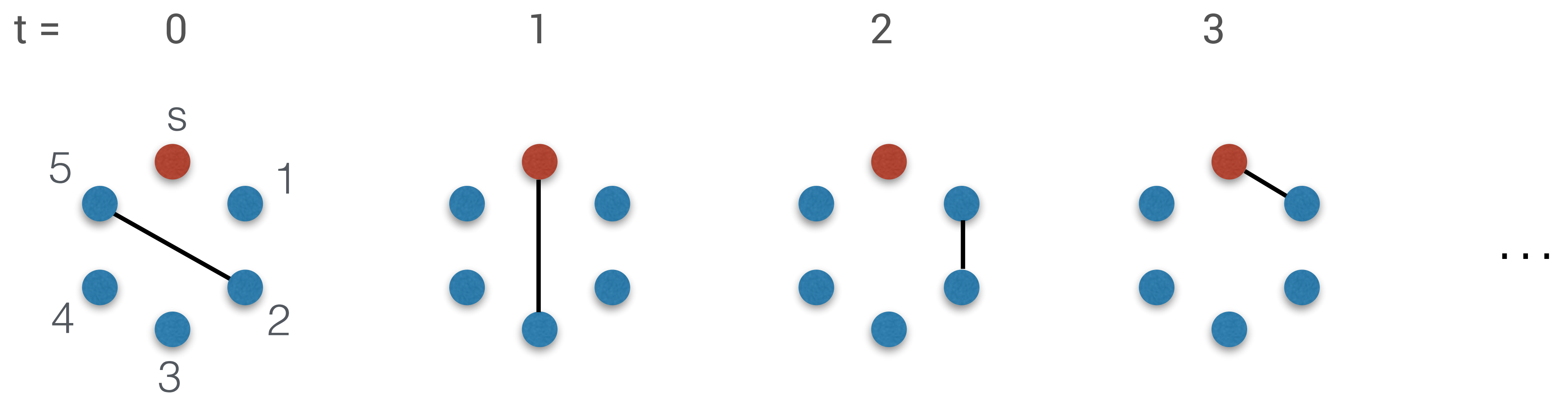
Distributed Online Data Aggregation



Who generates the sequence? An adversary

Three adversaries:

Distributed Online Data Aggregation

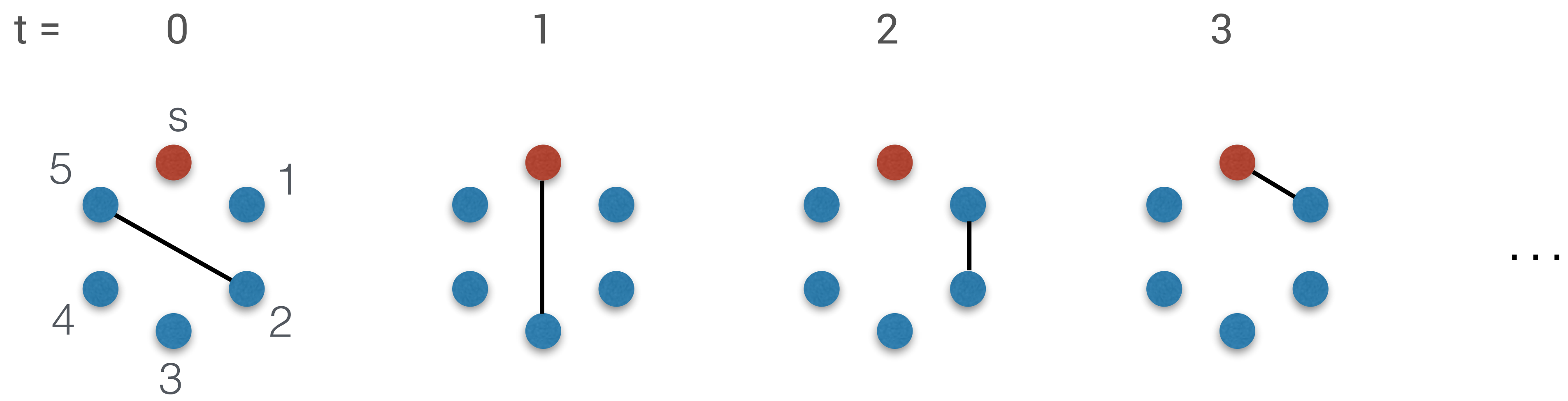


Who generates the sequence? An adversary

Three adversaries:

- **Online Adaptive**: generates the next interaction based on what happened in the past

Distributed Online Data Aggregation



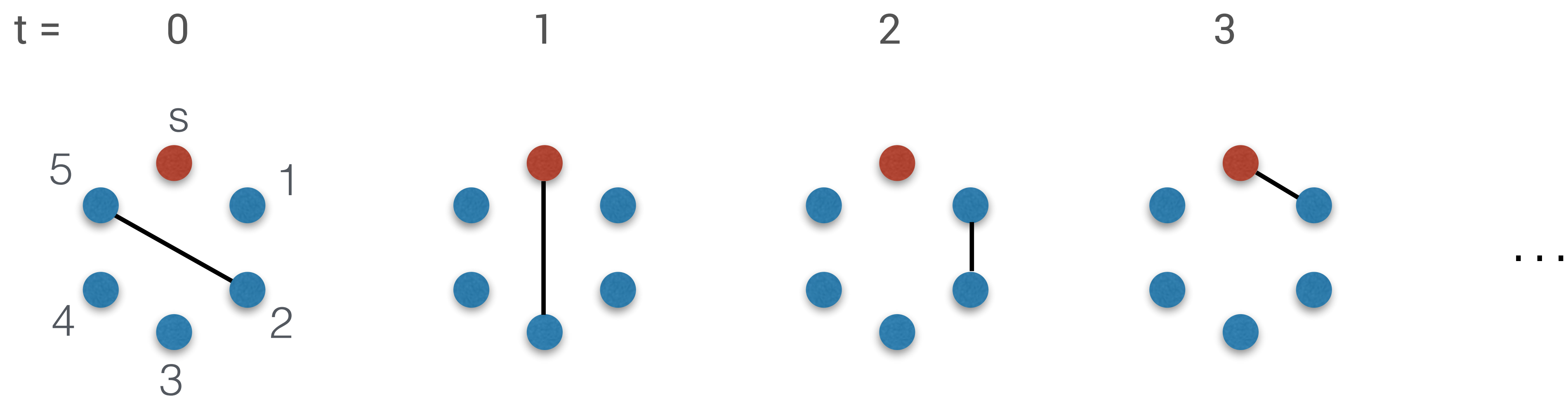
Who generates the sequence? An adversary

Three adversaries:

- **Online Adaptive**: generates the next interaction based on what happened in the past

- **Oblivious**: generates the sequence before the execution of the algorithm

Distributed Online Data Aggregation



Who generates the sequence? An adversary

Three adversaries:

- **Online Adaptive**: generates the next interaction based on what happened in the past

- **Oblivious**: generates the sequence before the execution of the algorithm

- **Randomized**: each interaction is chosen uniformly at random.

Impossibility Results - Online Adaptive Adversary

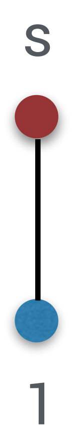
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction

Let A be a DODA

Impossibility Results - Online Adaptive Adversary

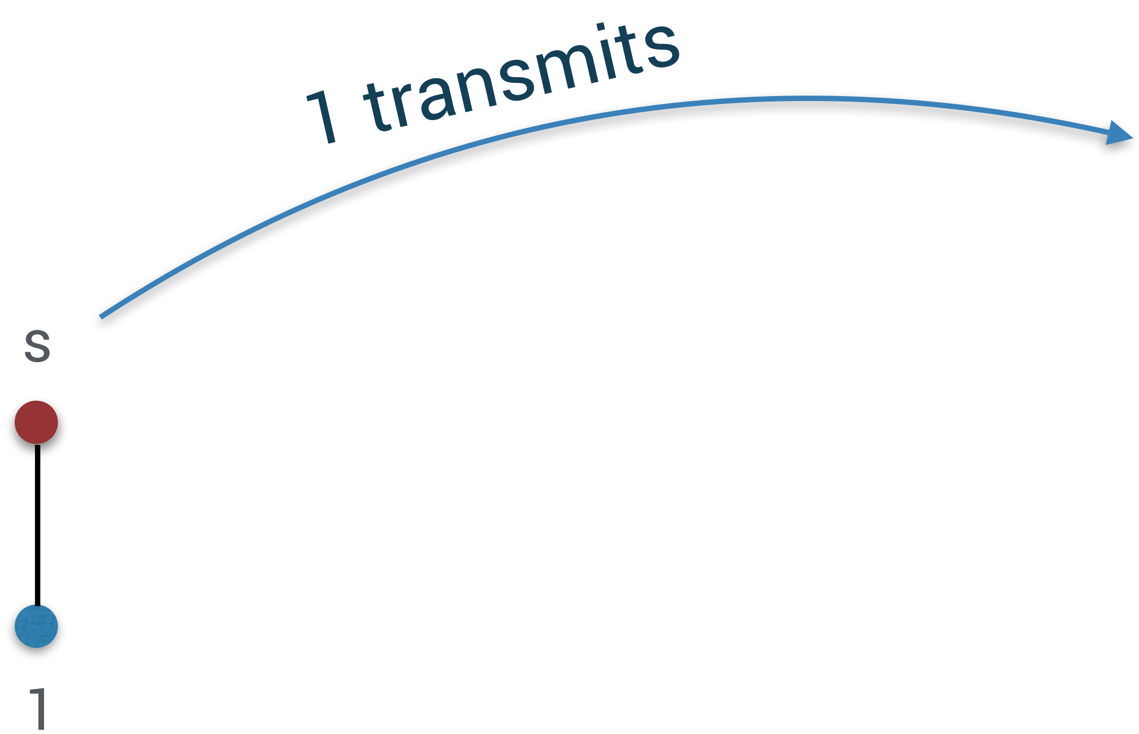
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction

Let A be a DODA



Impossibility Results - Online Adaptive Adversary

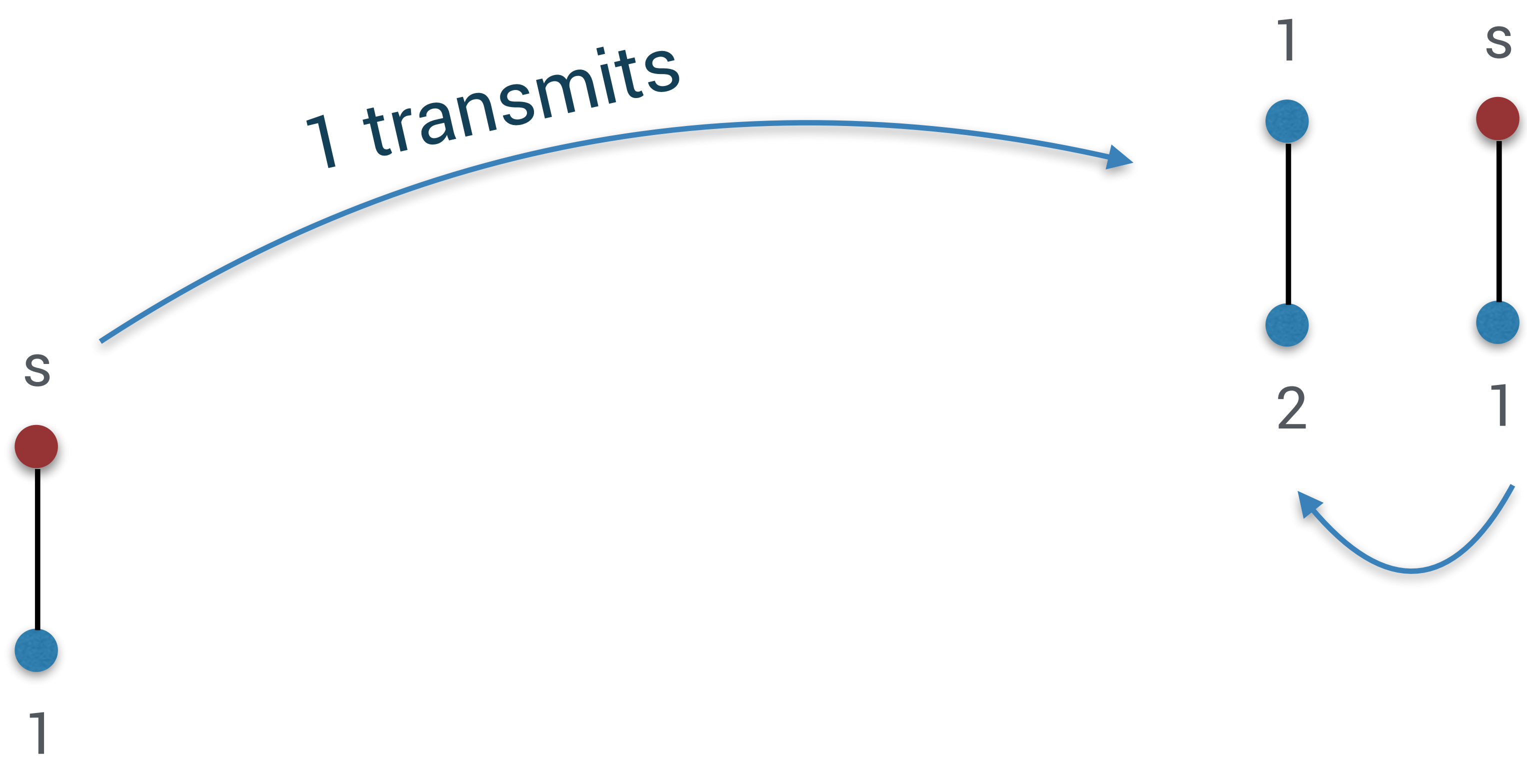
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

Impossibility Results - Online Adaptive Adversary

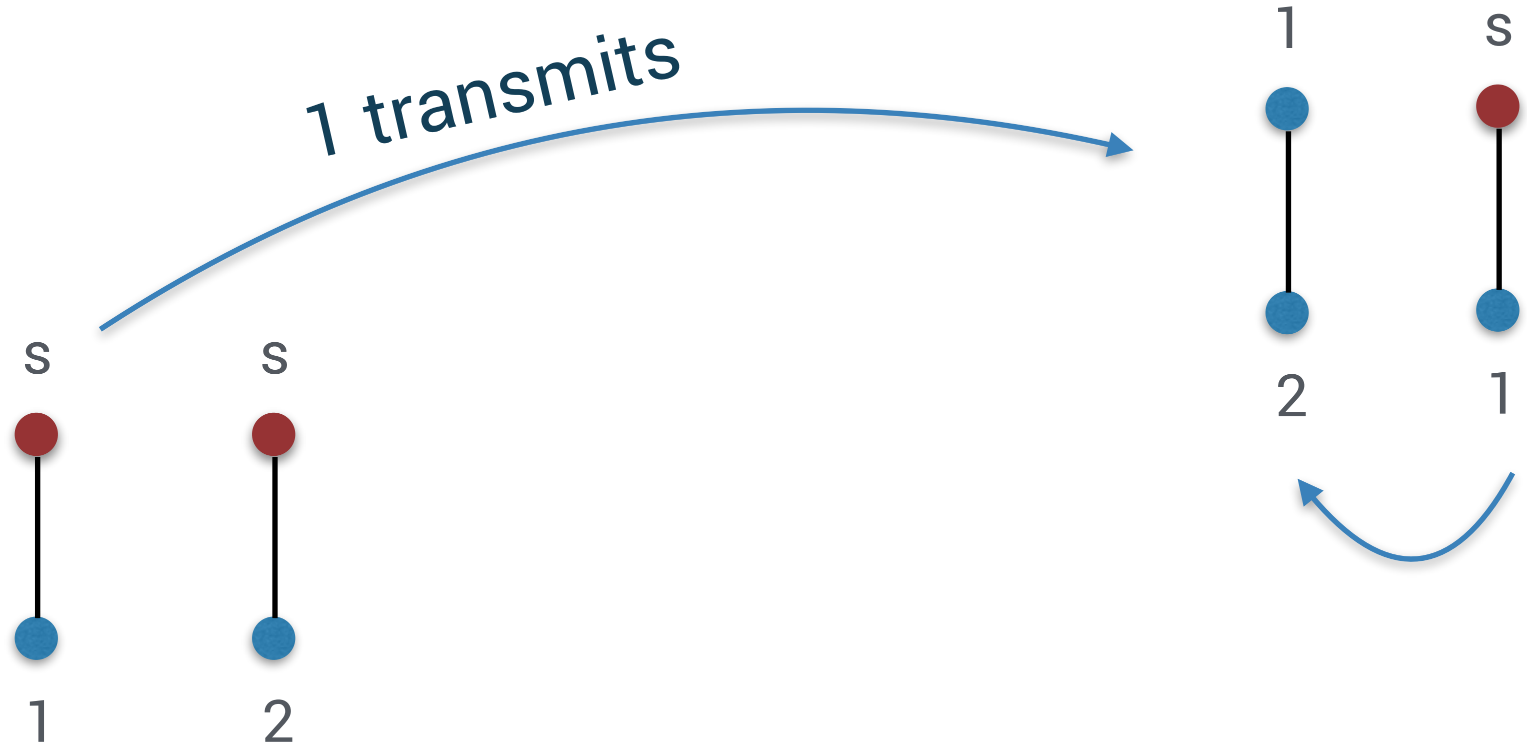
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

Impossibility Results - Online Adaptive Adversary

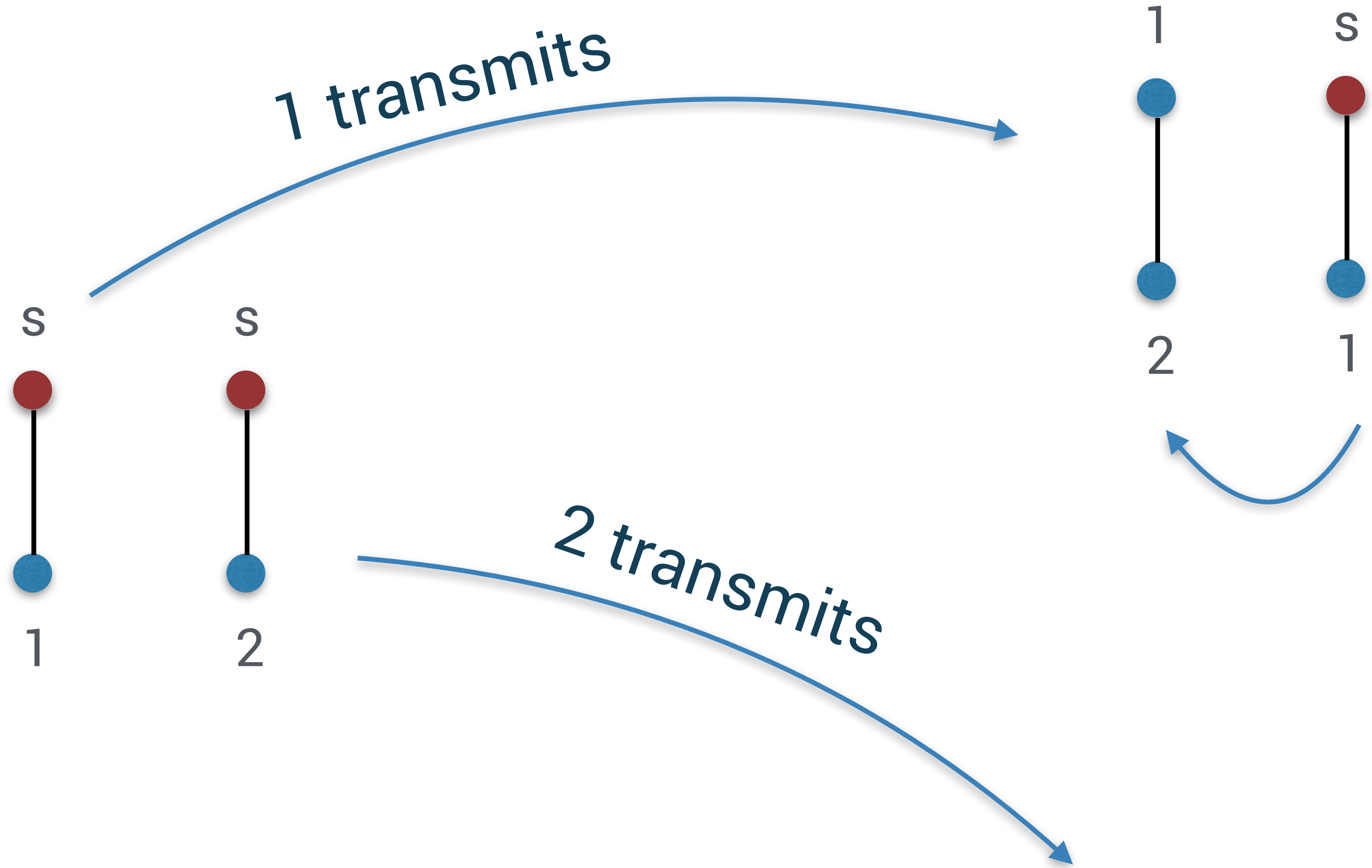
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

Impossibility Results - Online Adaptive Adversary

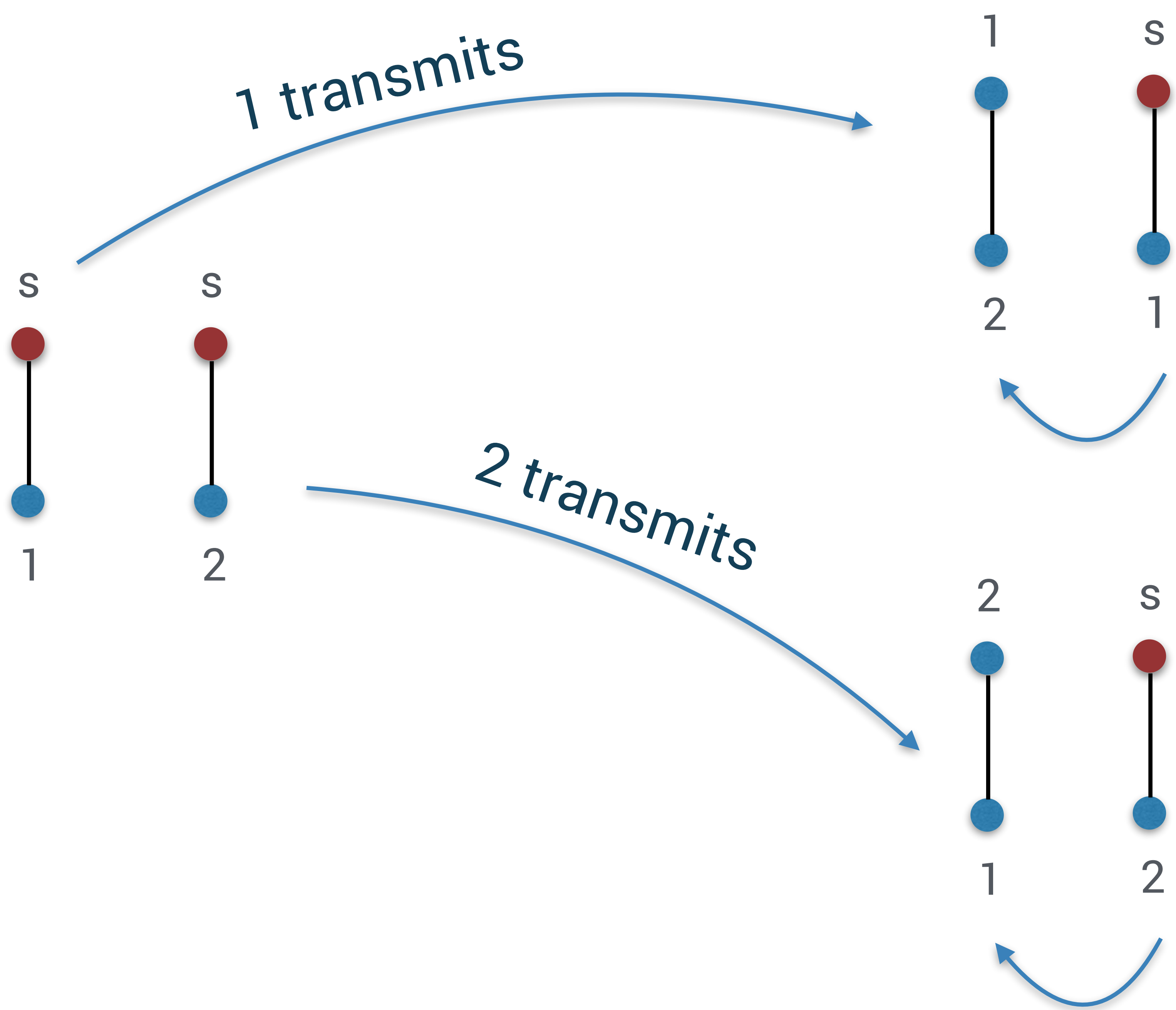
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

Impossibility Results - Online Adaptive Adversary

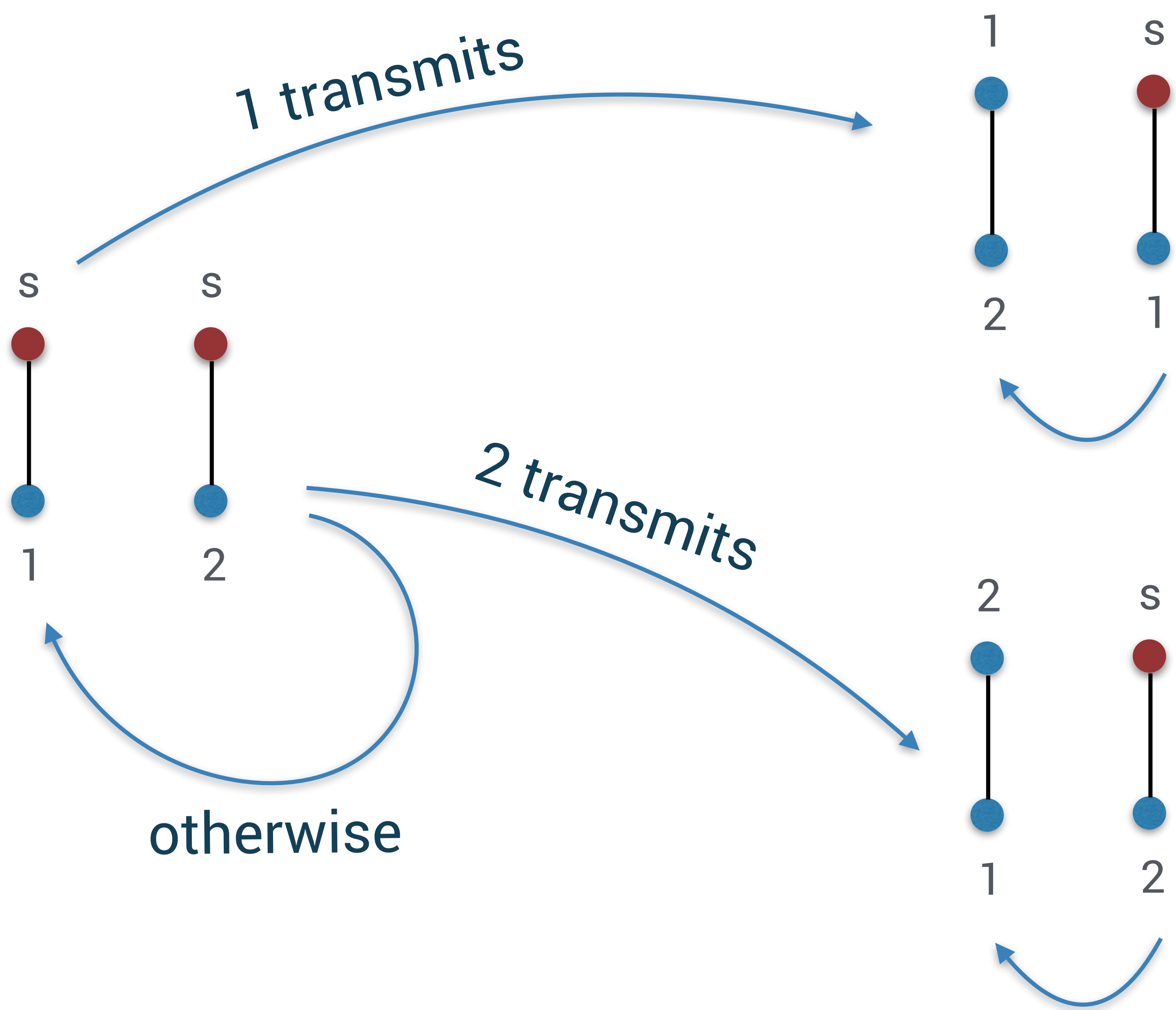
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

Impossibility Results - Online Adaptive Adversary

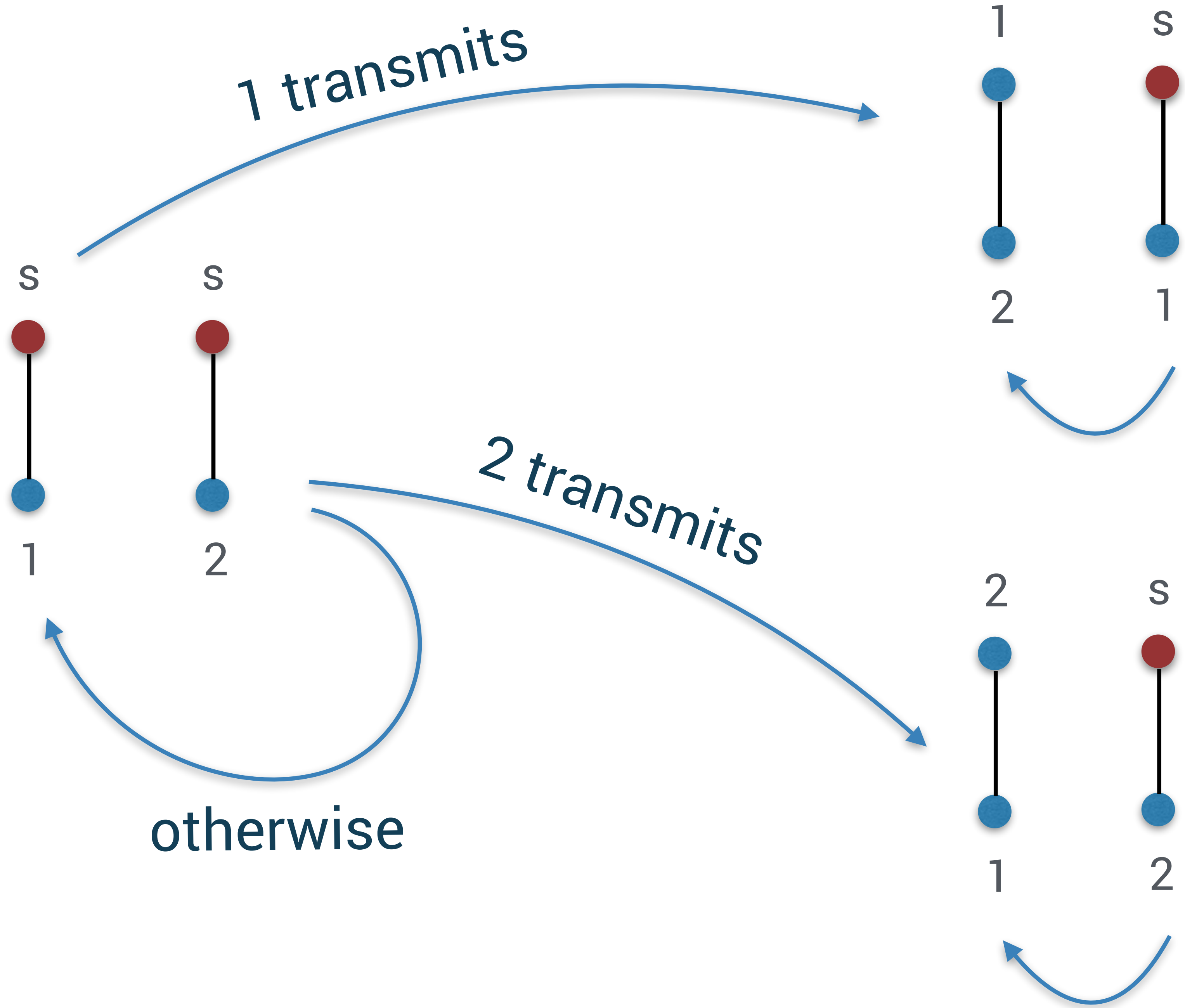
- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

Impossibility Results - Online Adaptive Adversary

- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction

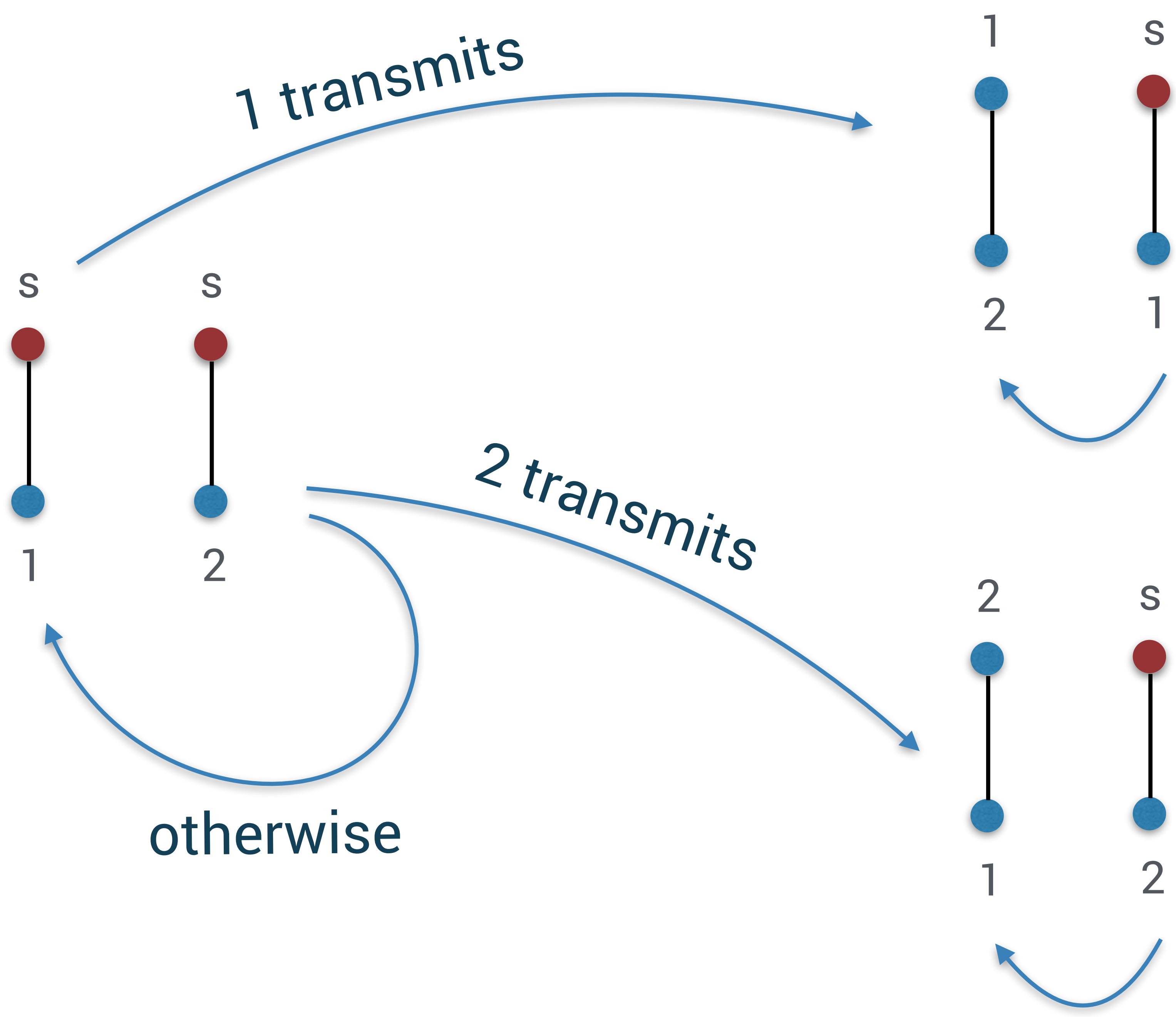


Let A be a DODA

A never terminates

Impossibility Results - Online Adaptive Adversary

- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



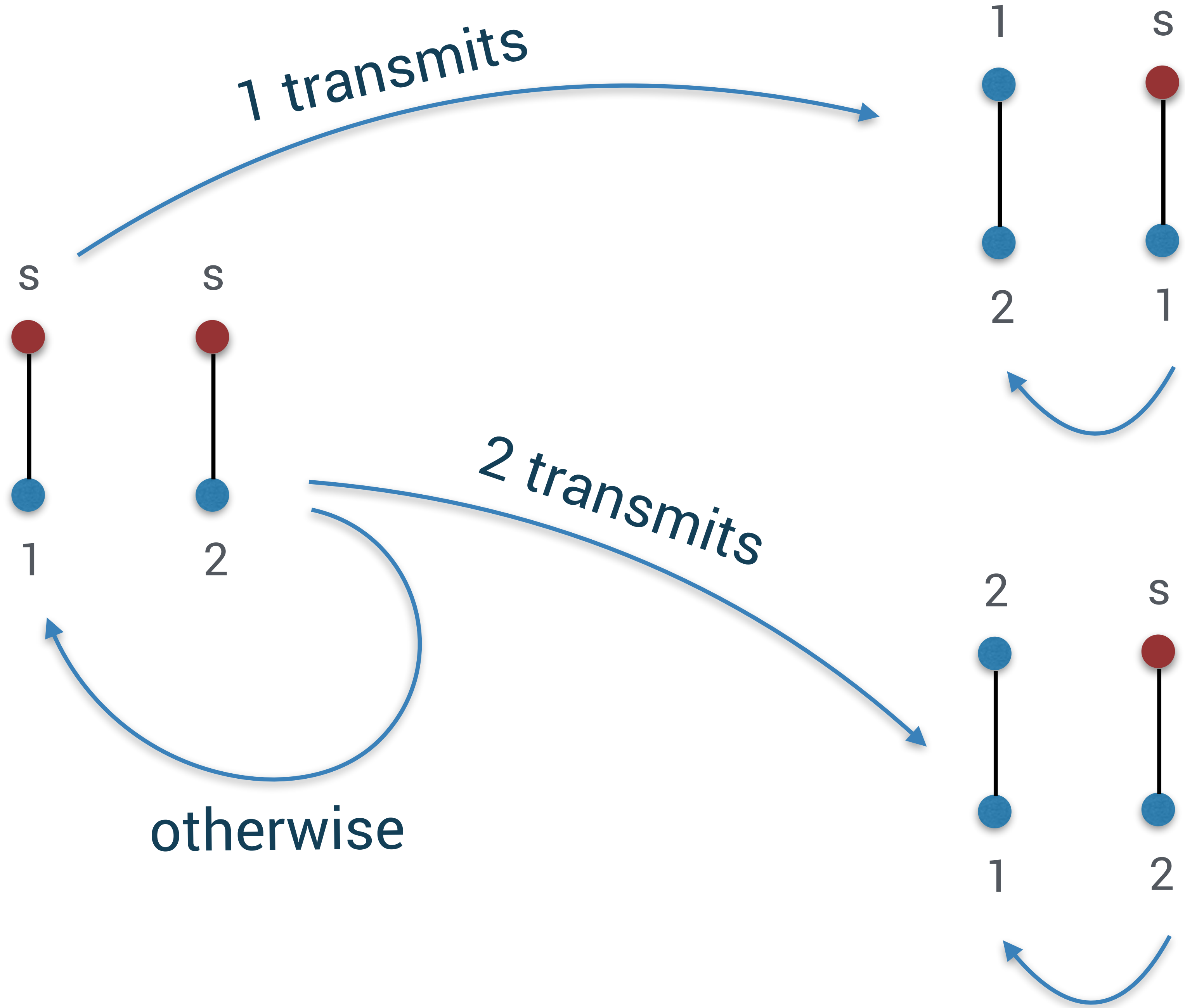
Let A be a DODA

A never terminates

Starting from any time t, the aggregation is always possible, so:

Impossibility Results - Online Adaptive Adversary

- **Online Adaptive**: generates the next interaction based on what the algorithm decides in the current interaction



Let A be a DODA

A never terminates

Starting from any time t, the aggregation is always possible, so:

$cost(A) = \infty$

Impossibility Results - Online Adaptive Adversary

Impossibility Results - Online Adaptive Adversary

Theorem: If k transmissions are allowed per node, there exists an online adaptive adversary that generates for every DODA A a sequence I such that $cost_I(A) = \infty$

Impossibility Results - Online Adaptive Adversary

Theorem: If k transmissions are allowed per node, there exists an online adaptive adversary that generates for every DODA A a sequence I such that $cost_I(A) = \infty$

Corollary: If k transmissions are allowed per node, there exists an **oblivious** adversary that generates for every **deterministic** DODA A a sequence I such that $cost_I(A) = \infty$

Impossibility Results - Online Adaptive Adversary

Theorem: If k transmissions are allowed per node, there exists an online adaptive adversary that generates for every DODA A a sequence I such that $cost_I(A) = \infty$

Corollary: If k transmissions are allowed per node, there exists an **oblivious** adversary that generates for every **deterministic** DODA A a sequence I such that $cost_I(A) = \infty$

What about randomized DODA algorithms against an oblivious adversary?

Impossibility Results - Oblivious Adversary

What about randomized DODA algorithms
against an oblivious adversary?

Theorem: If k transmissions are allowed per node, there exists an oblivious adversary that generates for every **oblivious** DODA A a sequence I such that $cost_I(A) = \infty$ with high probability

Impossibility Results - Oblivious Adversary

What about randomized DODA algorithms
against an oblivious adversary?

Theorem: If k transmissions are allowed per node, there exists an oblivious adversary that generates for every **oblivious** DODA A a sequence I such that $cost_I(A) = \infty$ with high probability

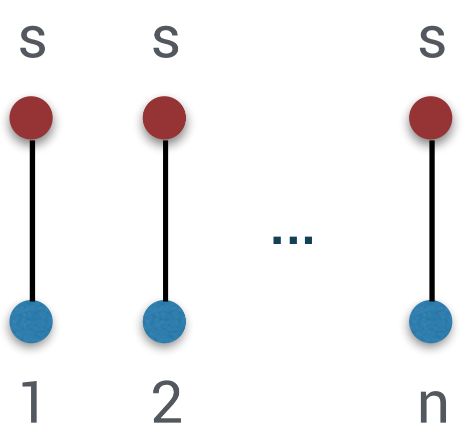
proof: not trivial

Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

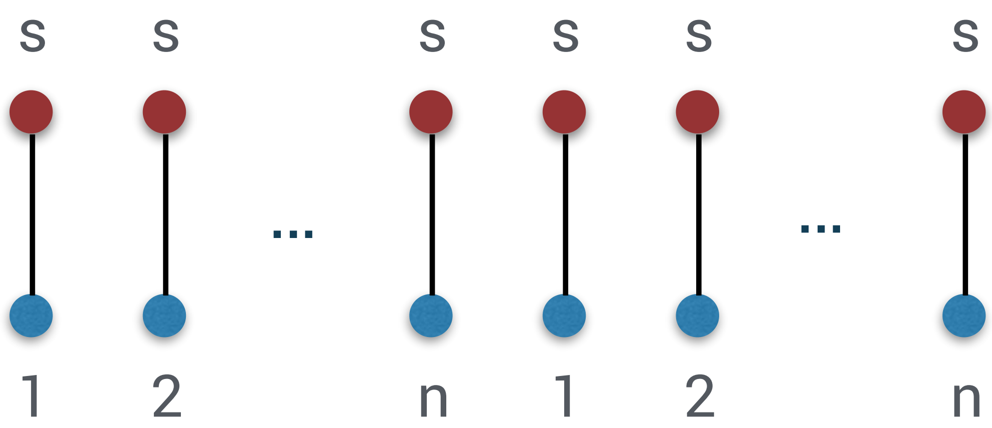
Impossibility Results - Oblivious Adversary

sketch of proof, for k=1



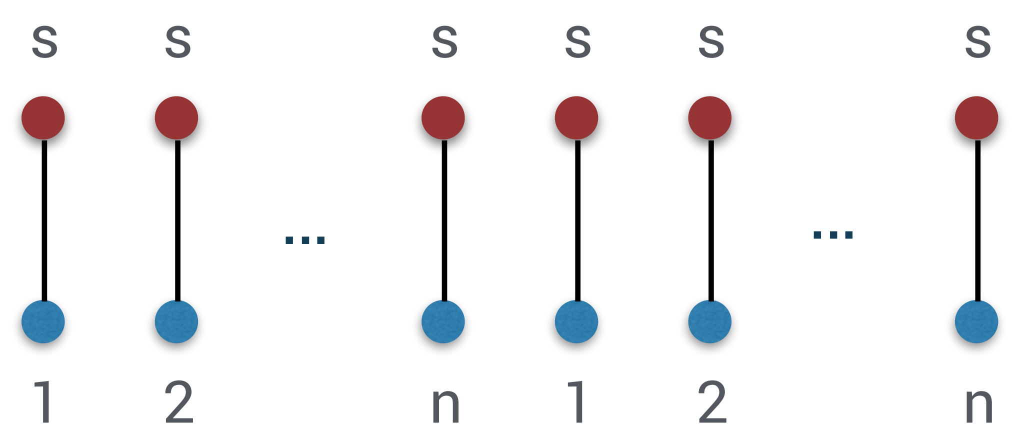
Impossibility Results - Oblivious Adversary

sketch of proof, for k=1



Impossibility Results - Oblivious Adversary

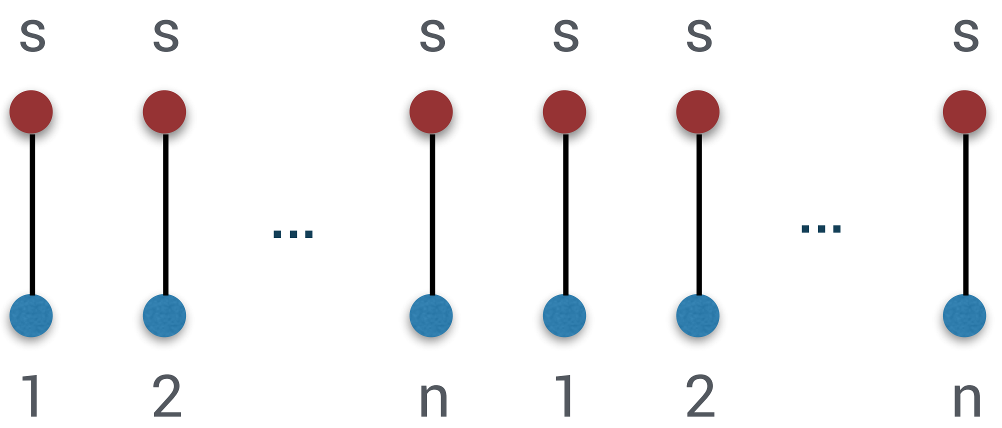
sketch of proof, for k=1



P_t = probability that no node transmits before time t

Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

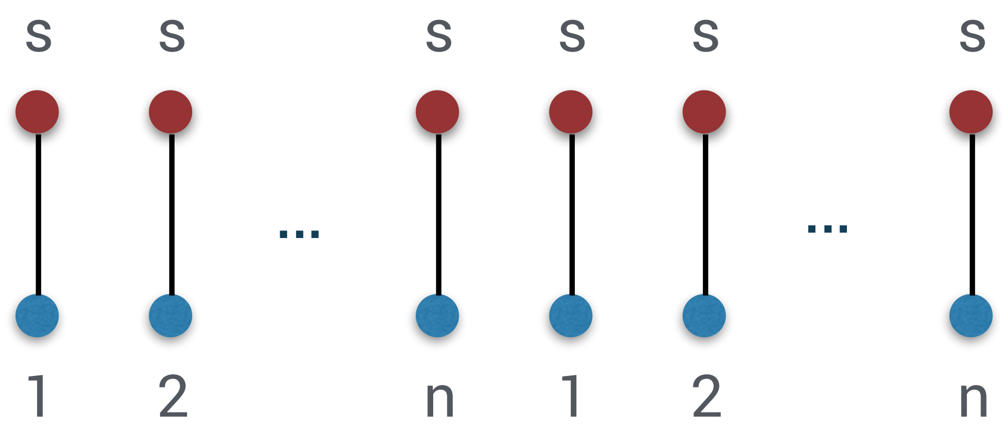


$P_t =$ probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

Impossibility Results - Oblivious Adversary

sketch of proof, for k=1



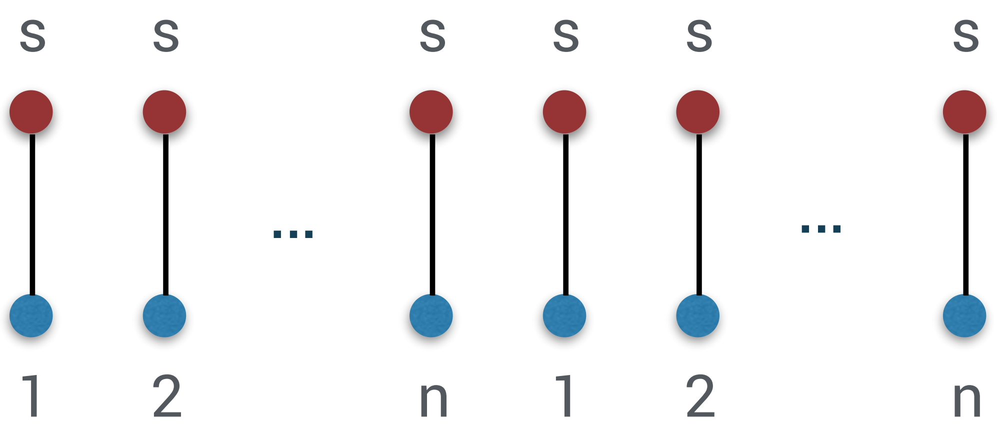
P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

if $P_t > 1/n$ for all t, then it's over

Impossibility Results - Oblivious Adversary

sketch of proof, for k=1



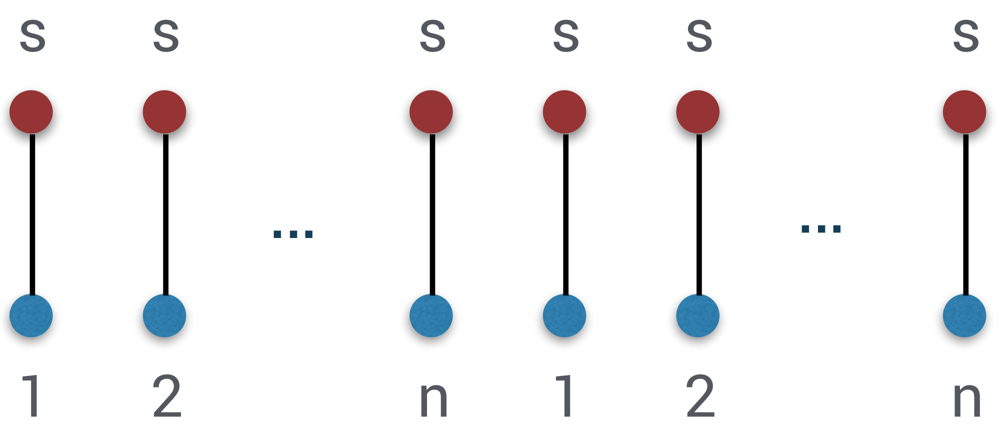
P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

Impossibility Results - Oblivious Adversary

sketch of proof, for k=1



P_t = probability that no node transmits before time t

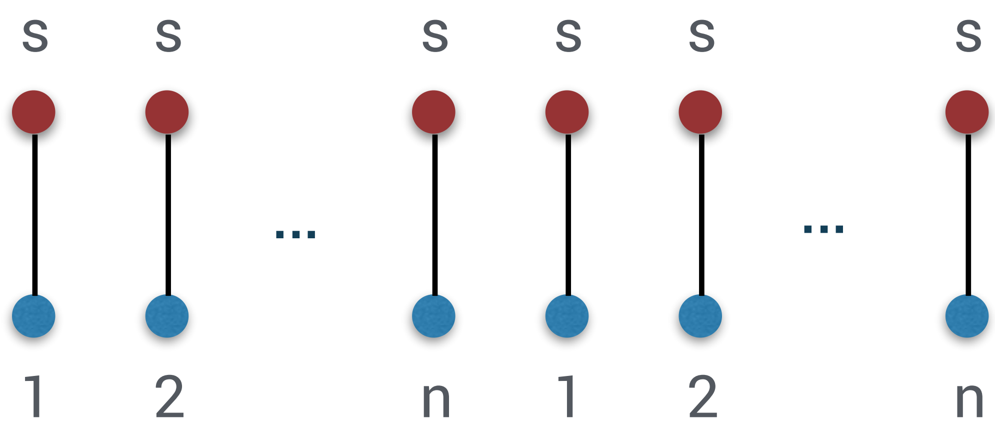
lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

- $P_{t0-1} > 1/n$
- $P_{t0} \leq 1/n$

Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

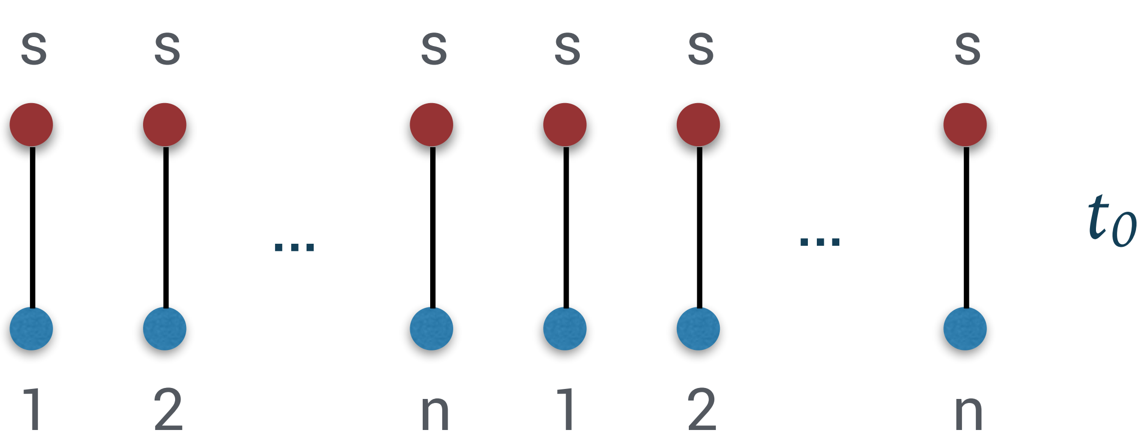


P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

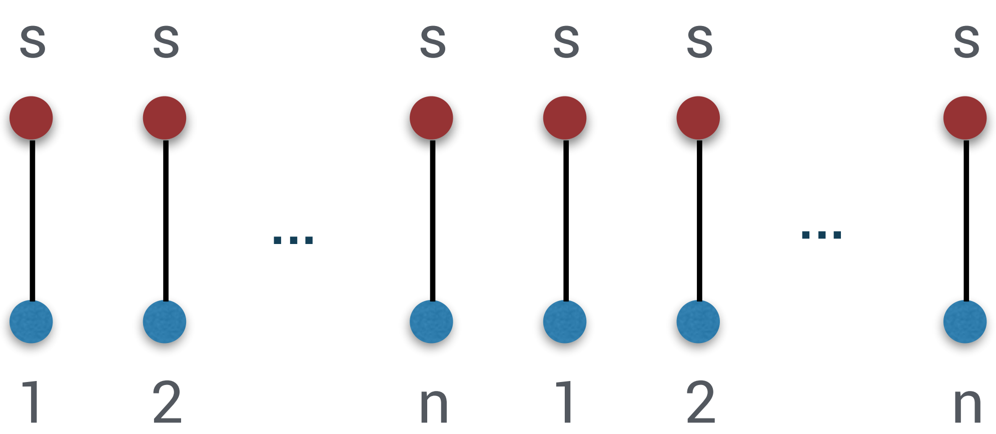
if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

- $P_{t0\,-1} > 1/n$
- $P_{t0} \leq 1/n$



Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

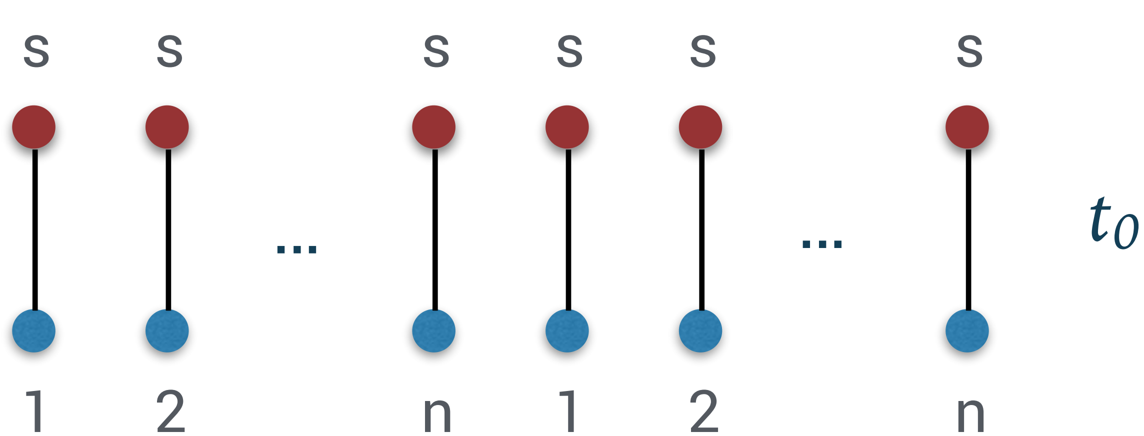


P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

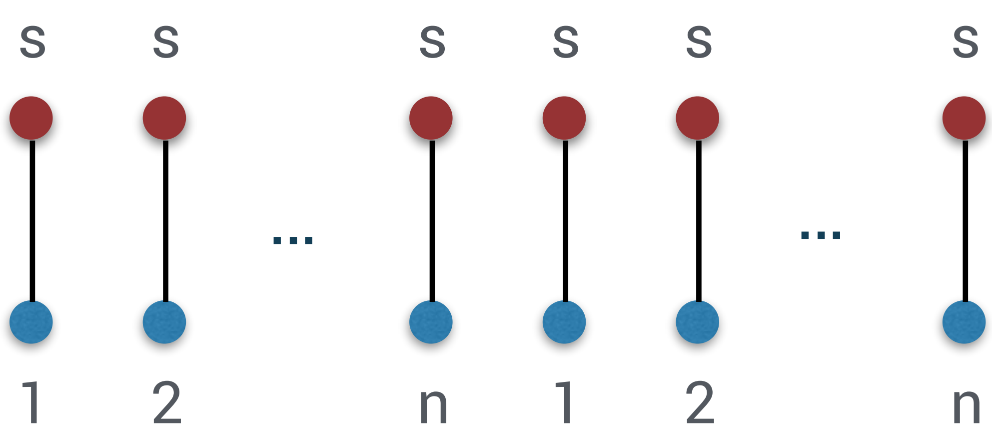
if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

- $P_{t0\,-1} > 1/n$: one node u **has not** transmitted w.h.p.
- $P_{t0} \leq 1/n$



Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

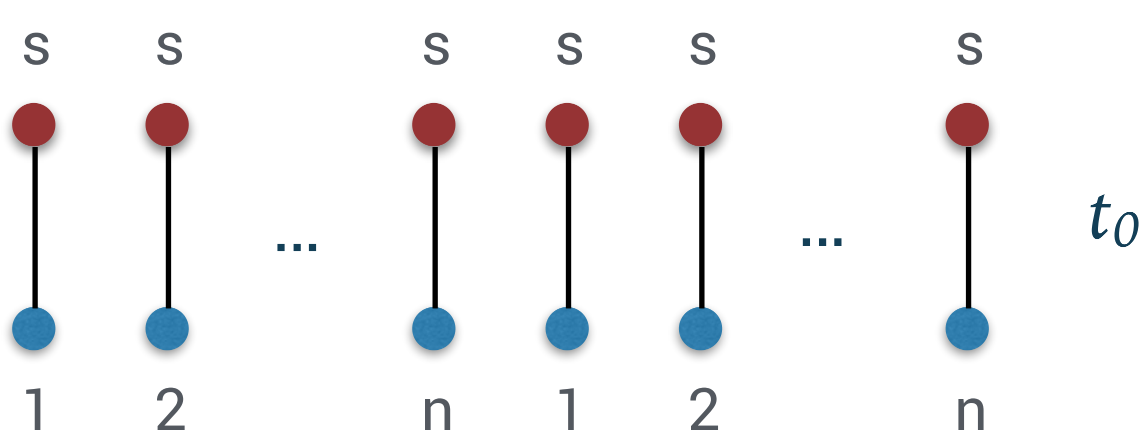


P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

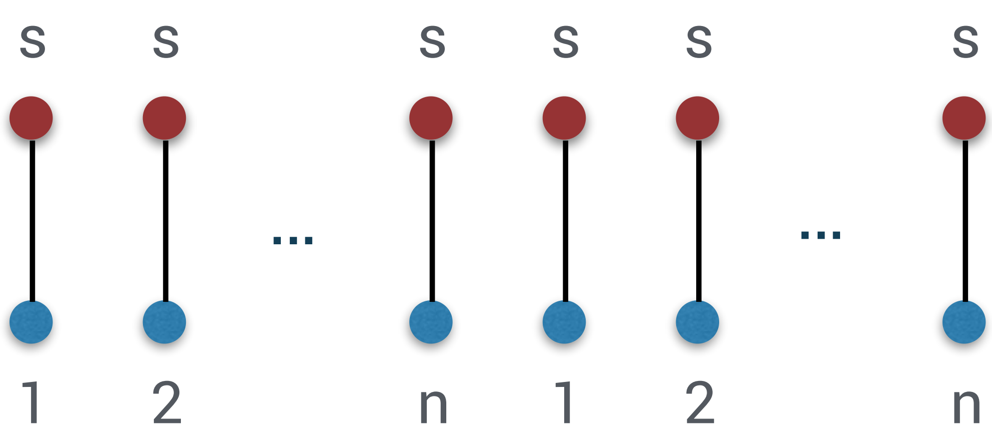
if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

- $P_{t0-1} > 1/n$: one node u **has not** transmitted w.h.p.
- $P_{t0} \leq 1/n$: one node **has** transmitted w.h.p.



Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

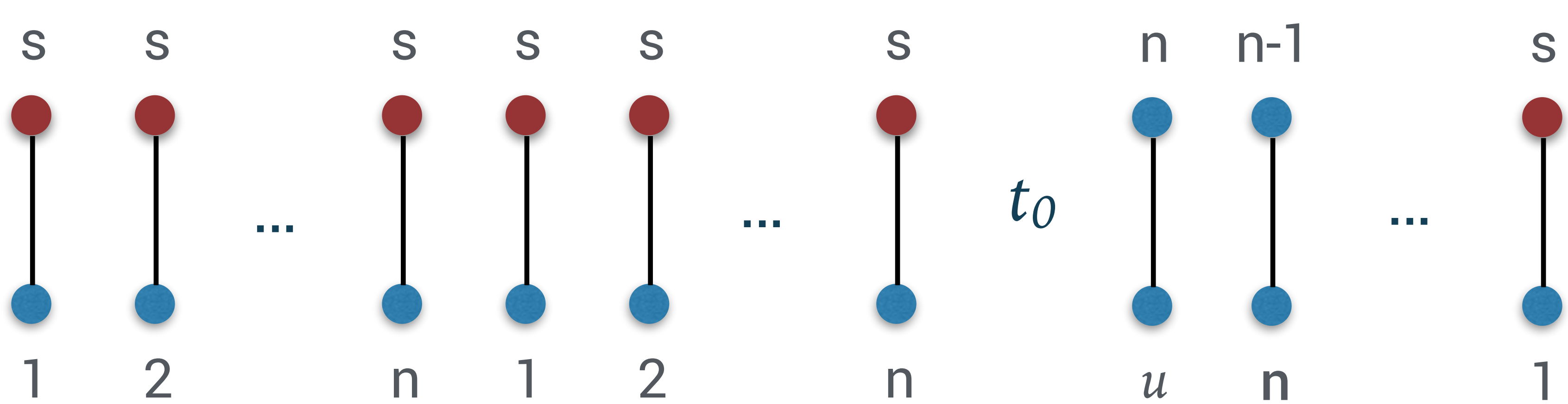


P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

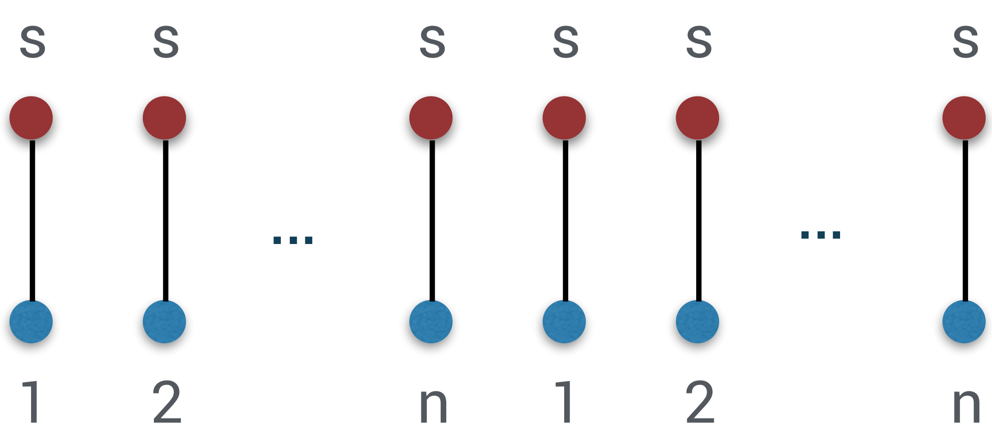
if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

- $P_{t0\,-1} > 1/n$: one node u **has not** transmitted w.h.p.
- $P_{t0} \leq 1/n$: one node **has** transmitted w.h.p.



Impossibility Results - Oblivious Adversary

sketch of proof, for k=1

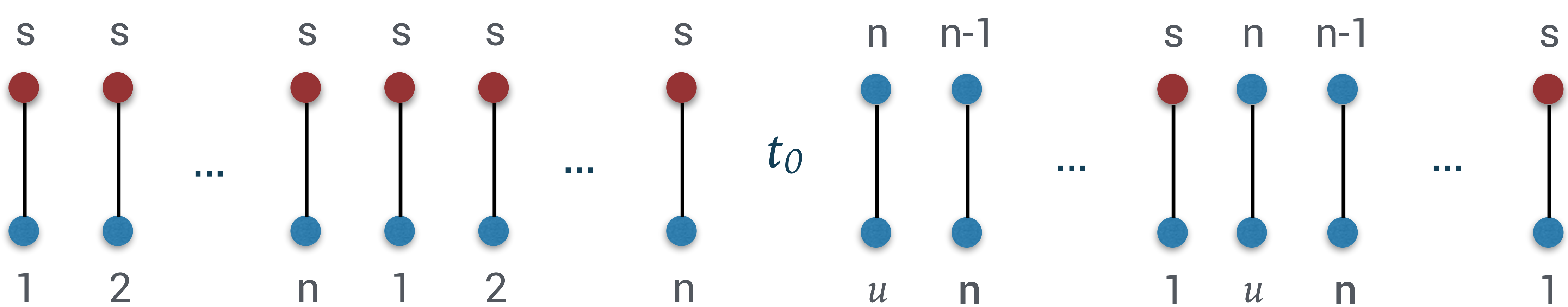


P_t = probability that no node transmits before time t

lemma: if $P_t > 1/n$ then there exists a node u that has not transmitted w.h.p.

if $P_t > 1/n$ for all t, then it's over, otherwise there is a time t_0 such that:

- $P_{t0\,-1} > 1/n$: one node u **has not** transmitted w.h.p.
- $P_{t0} \leq 1/n$: one node **has** transmitted w.h.p.



Impossibility Results - Oblivious Adversary

sketch of proof, for k>1

I^{k-1}

I^{k-1}

$\vdots$

I^{k-1}

Each group of nodes acts like a node
that can transmit only once

require n^k nodes

Impossibility Results - Oblivious Adversary

What about randomized DODA algorithms against oblivious adversary?

Theorem: If k transmissions are allowed per node, there exists an oblivious adversary that generate for every **oblivious** DODA A a sequence I such that $cost_I(A) = \infty$ with high probability

Impossibility Results - Oblivious Adversary

What about randomized DODA algorithms against oblivious adversary?

Theorem: If k transmissions are allowed per node, there exists an oblivious adversary that generate for every **oblivious** DODA A a sequence I such that $cost_I(A) = \infty$ with high probability

What about non-oblivious randomized DODA algorithms against oblivious adversary?

Impossibility Results - Oblivious Adversary

What about randomized DODA algorithms against oblivious adversary?

Theorem: If k transmissions are allowed per node, there exists an oblivious adversary that generate for every **oblivious** DODA A a sequence I such that $cost_I(A) = \infty$ with high probability

What about non-oblivious randomized DODA algorithms against oblivious adversary?

open question

DODA against randomized adversary

DODA against randomized adversary

If you know everything about the future,
it takes $O(nlog(n))$ interactions in average.

If you have no information about the future, always transmit is optimal.
It takes $O(n^2)$ interactions in average.

DODA against randomized adversary

DODA against randomized adversary

The $meetTime$ information: each node knows when will be its next interaction with the sink

DODA against randomized adversary

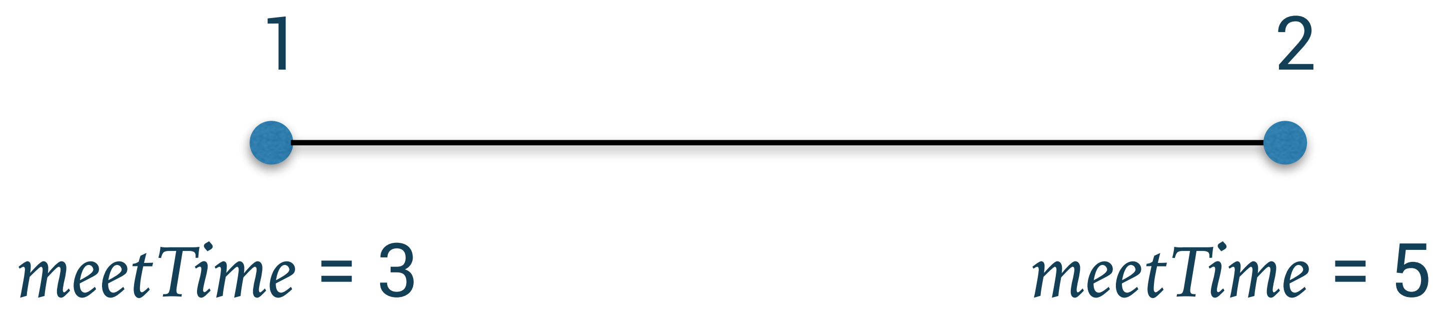
The *meetTime* information: each node knows when will be its next interaction with the sink

DODA against randomized adversary

Greedy Algorithm: **if** I meet the sink after the other node, **then** I transmit my data

DODA against randomized adversary

Greedy Algorithm: **if** I meet the sink after the other node, **then** I transmit my data



DODA against randomized adversary

Greedy Algorithm: **if** I meet the sink after the other node, **then** I transmit my data

DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has *meetTime* > τ,
then we apply Greedy Algorithm

DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$,
then we apply Greedy Algorithm

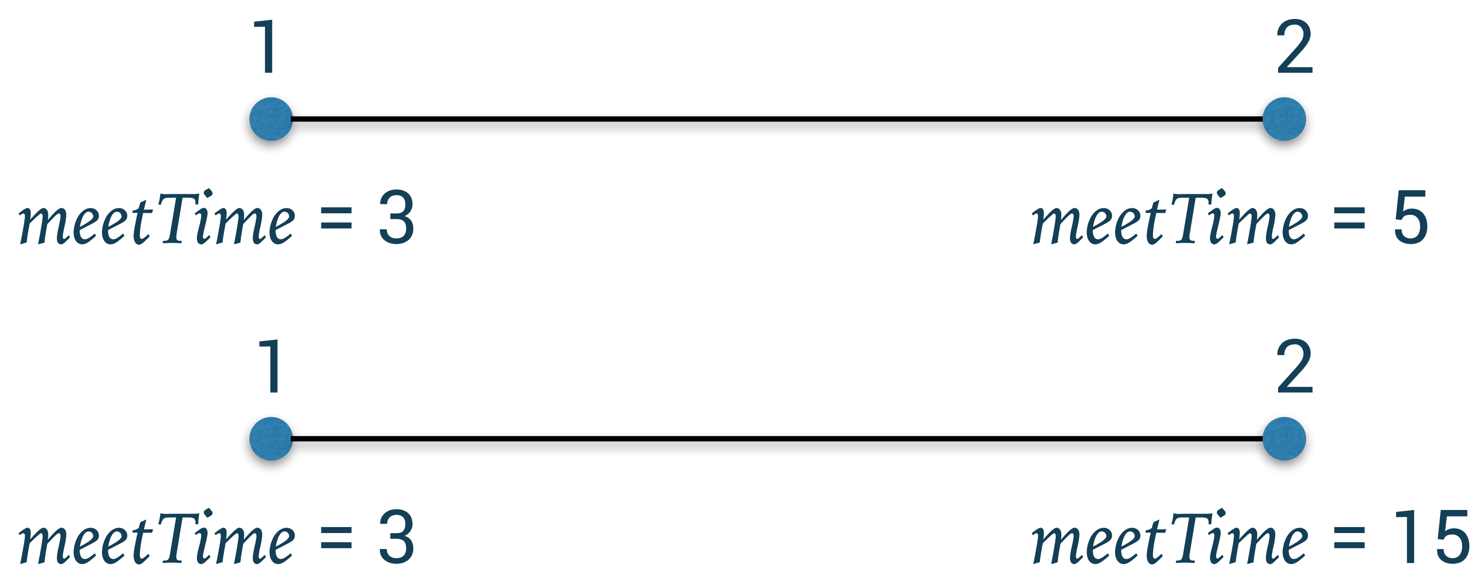
$\tau = 10$

$meetTime = 3$ $meetTime = 5$

DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has *meetTime* > τ, **then** we apply Greedy Algorithm

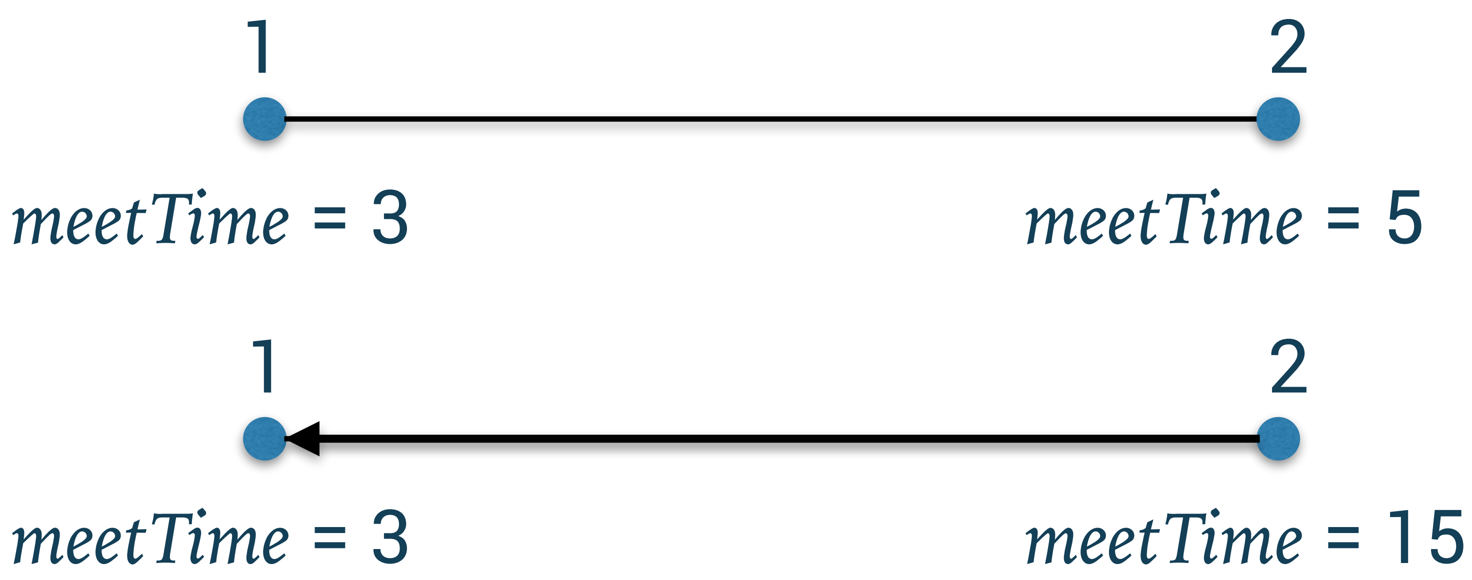
$\tau = 10$



DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$,
then we apply Greedy Algorithm

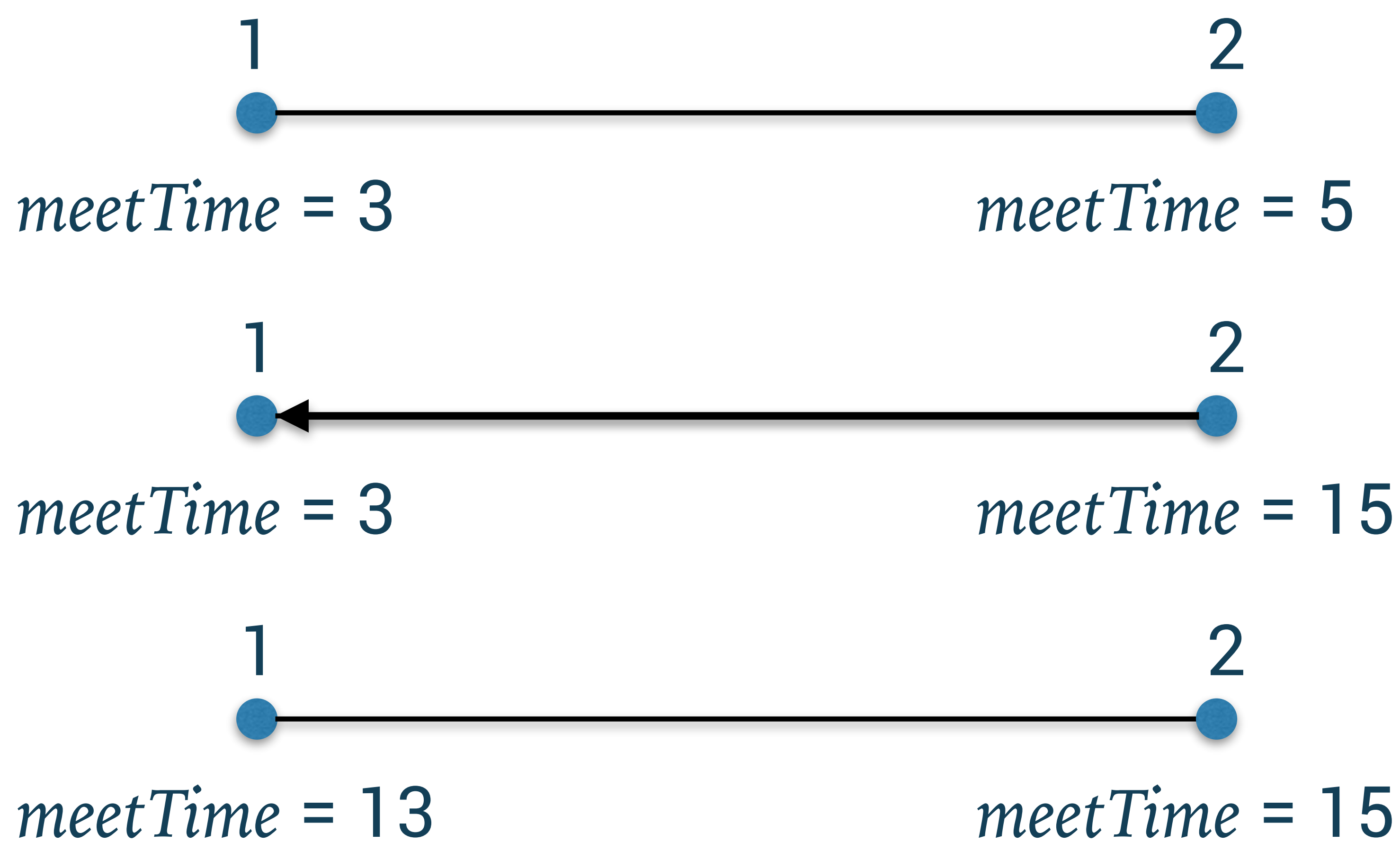
$\tau = 10$



DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$, **then** we apply Greedy Algorithm

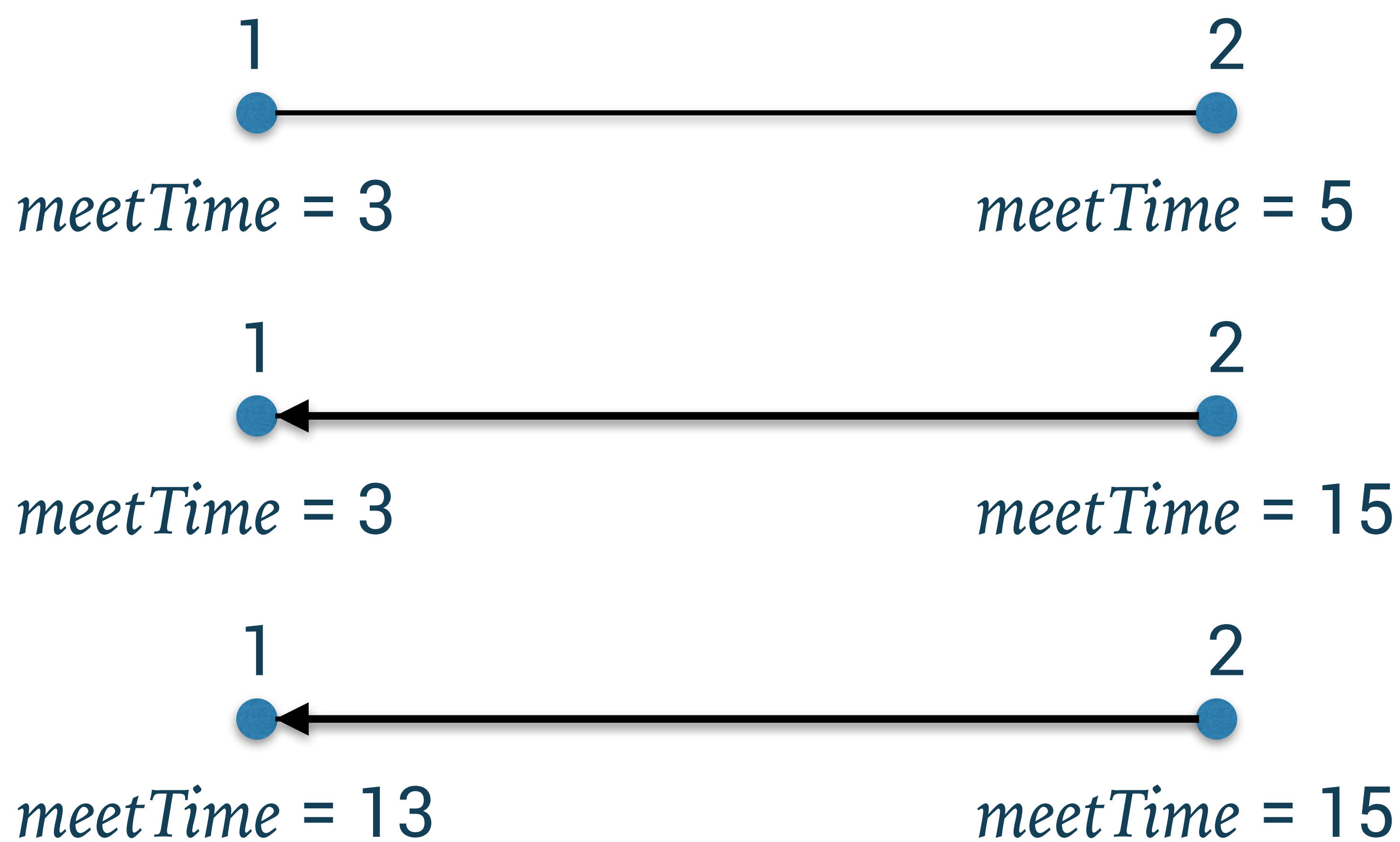
$\tau = 10$



DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime$ > τ, **then** we apply Greedy Algorithm

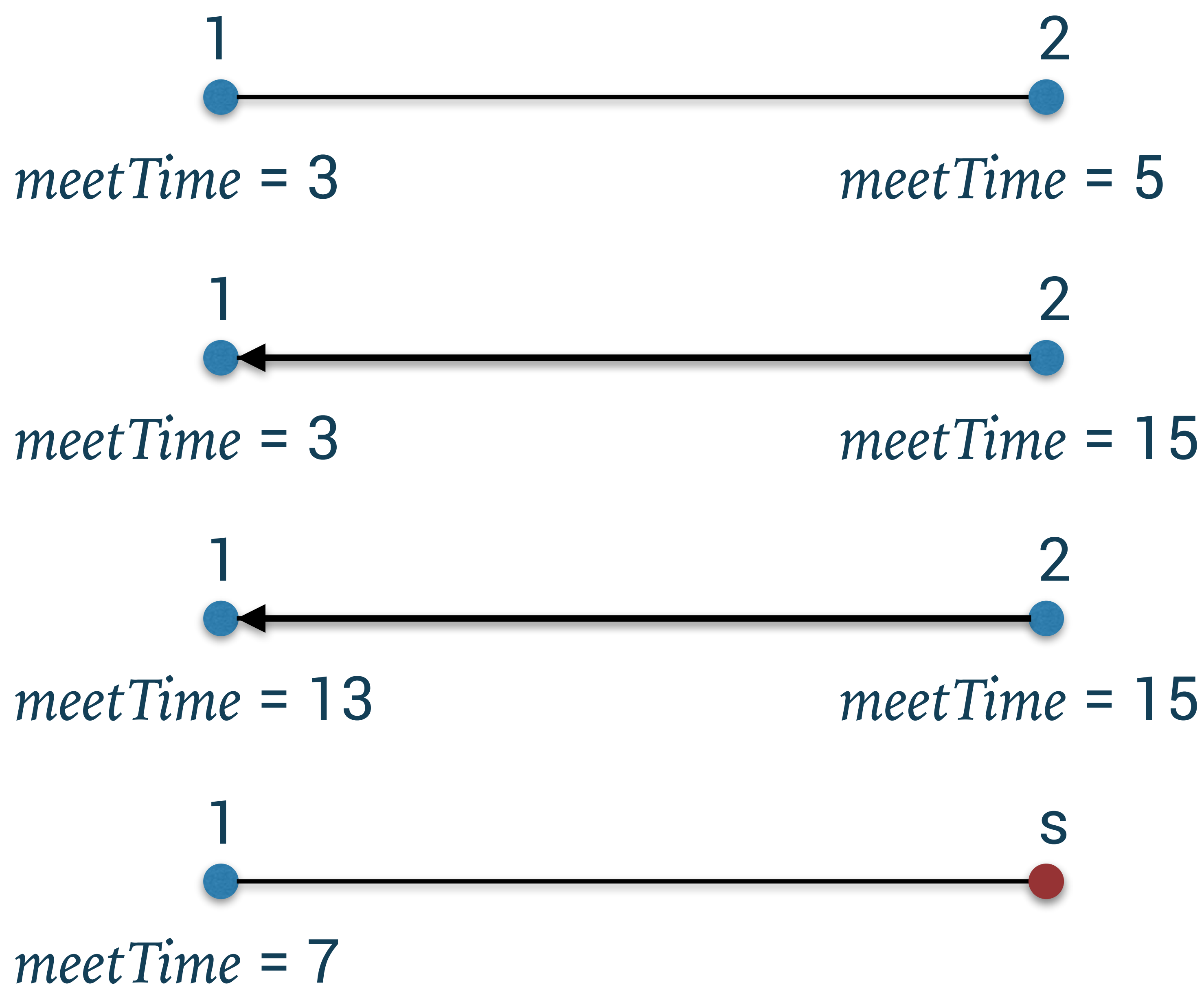
τ =10



DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$, **then** we apply Greedy Algorithm

$\tau = 10$



DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$,
then we apply Greedy Algorithm

$n\sqrt{n\ log(n)}$-Waiting Greedy Algorithm is optimal

DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$,
then we apply Greedy Algorithm

$n\sqrt{n \, log(n)}$-Waiting Greedy Algorithm is optimal

Without knowledge: $\Theta(n^2)$ interactions in average

DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$,
then we apply Greedy Algorithm

$n\sqrt{n\ log(n)}$-Waiting Greedy Algorithm is optimal

Without knowledge: $\Theta(n^2)$ interactions in average

With full knowledge: $\Theta(n\ log(n))$ interactions w.h.p.

DODA against randomized adversary

τ-Waiting Greedy Algorithm: **if** at least one of the nodes has $meetTime > \tau$,
then we apply Greedy Algorithm

$n\sqrt{n \log(n)}$-Waiting Greedy Algorithm is optimal

Without knowledge:	$\Theta(n^2)$	interactions in average
$meetTime$ information:	$\Theta(n\sqrt{n \log(n)})$	interactions w.h.p.
With full knowledge:	$\Theta(n \log(n))$	interactions w.h.p.

Conclusion

Conclusion

Online Data Aggregation:

- hard in general
- optimal algorithms exist in randomized networks

Conclusion

Online Data Aggregation:
- hard in general
- optimal algorithms exist in randomized networks

Perspective

Conclusion

Online Data Aggregation:

- hard in general
- optimal algorithms exist in randomized networks

Perspective

More realistic networks?

Conclusion

Online Data Aggregation:
- hard in general
- optimal algorithms exist in randomized networks

Perspective

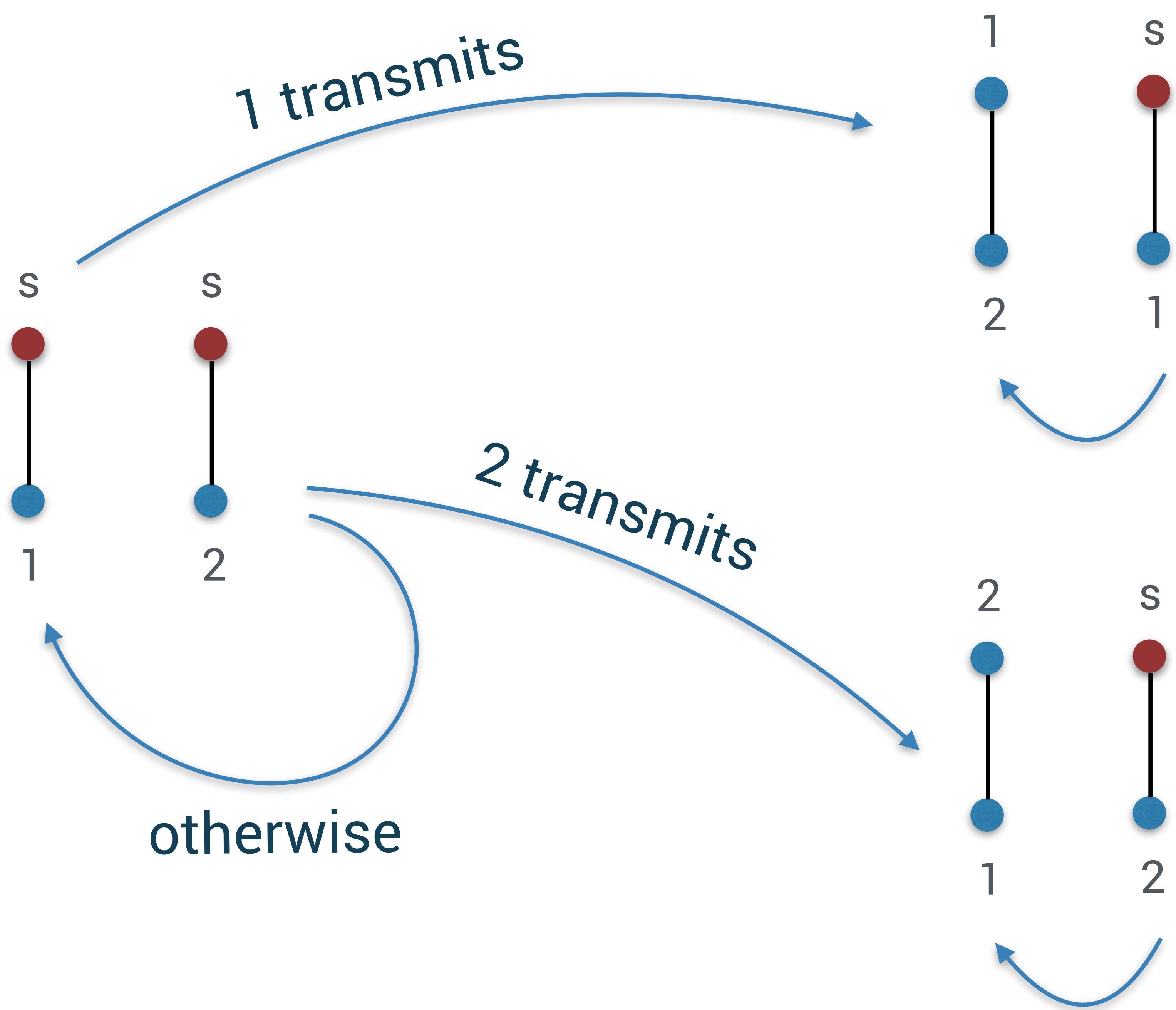
More realistic networks?

Thank you for your attention!

quentin.bramas@lip6.fr

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



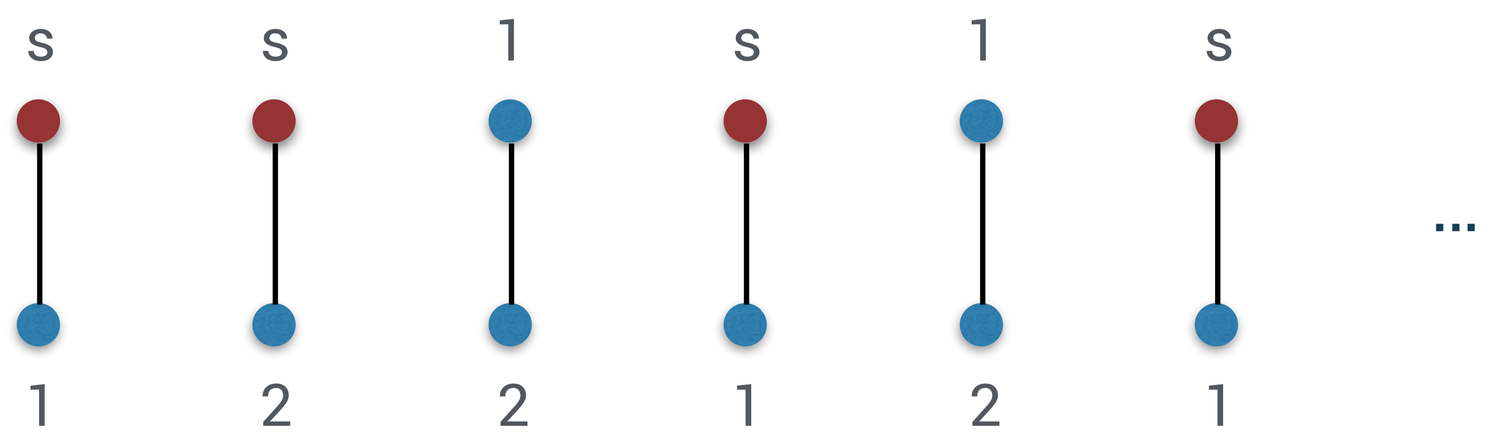
Let A be a DODA

Starting from any time t, the aggregation is always possible, so:

$$cost(A) = \infty$$

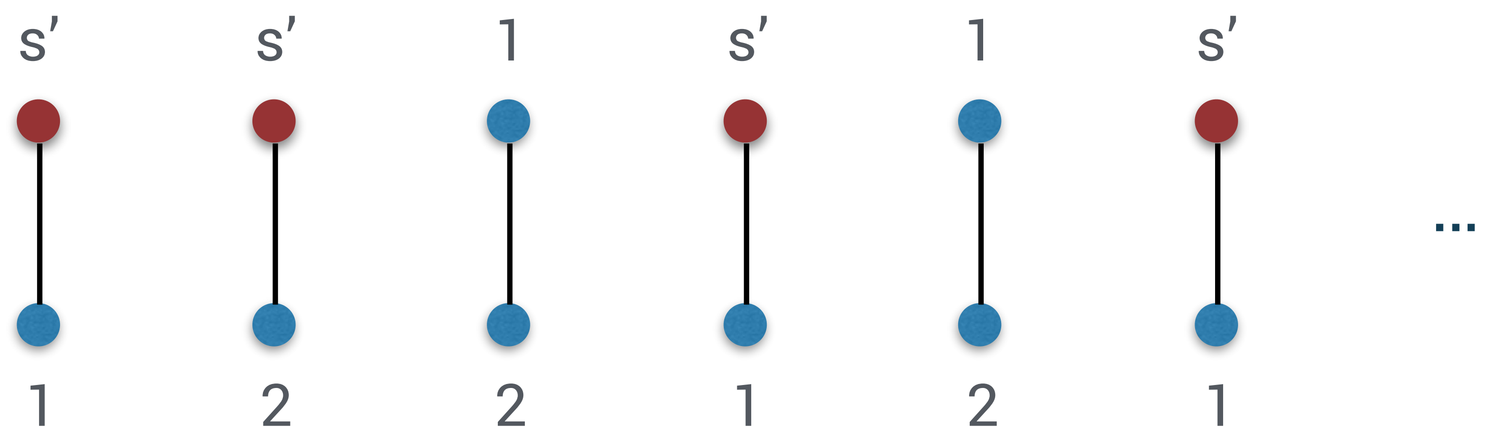
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



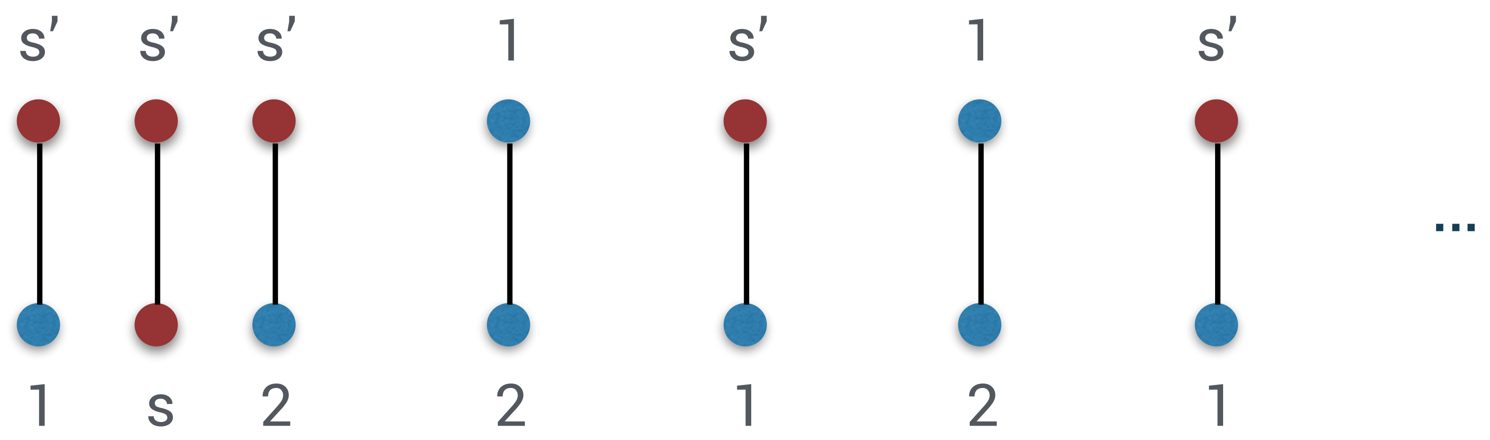
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



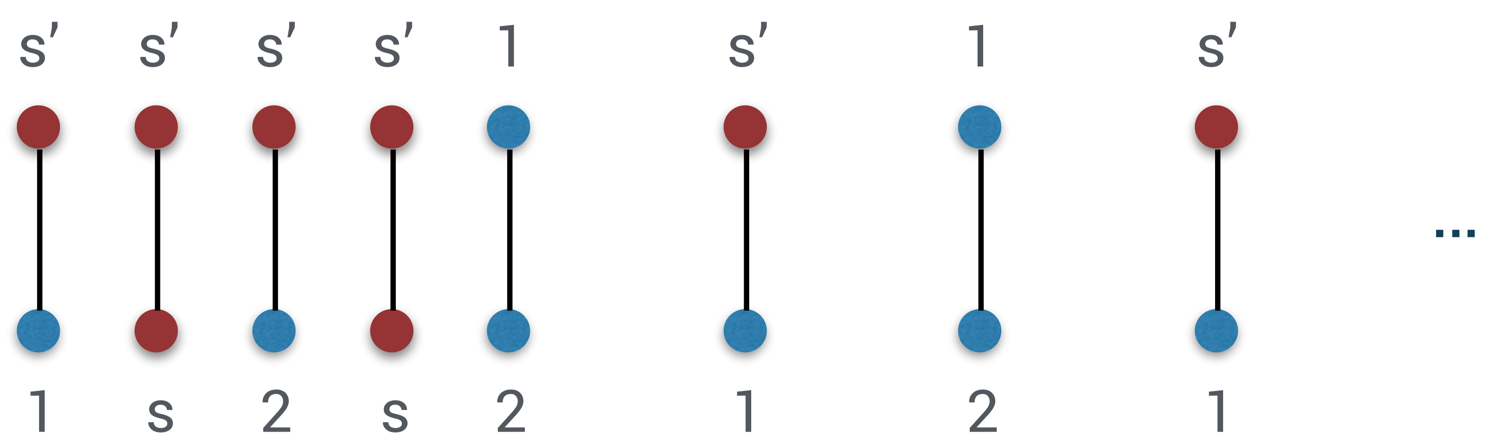
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



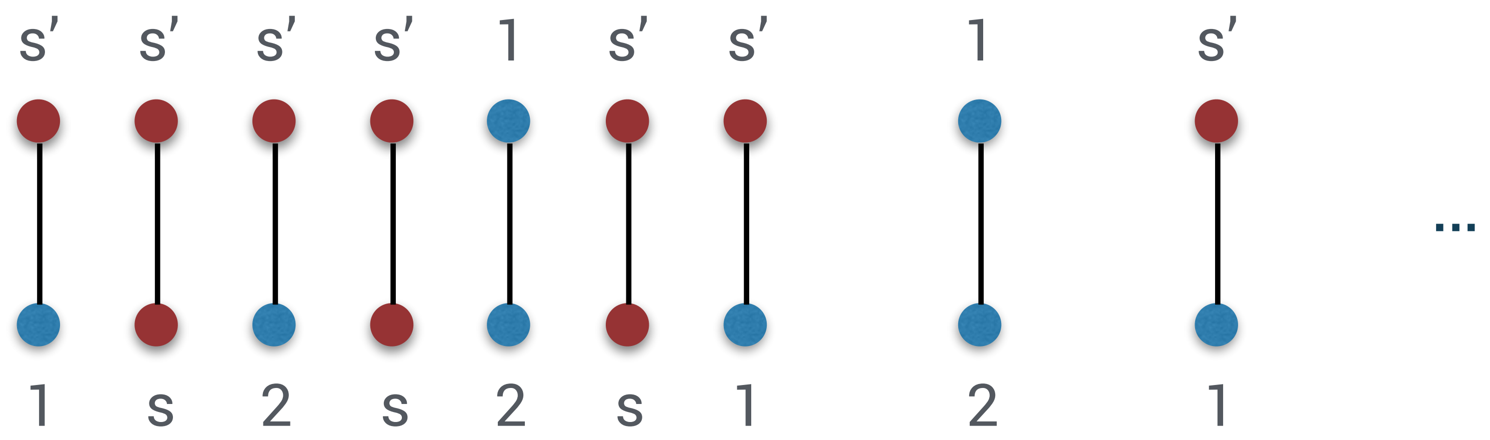
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



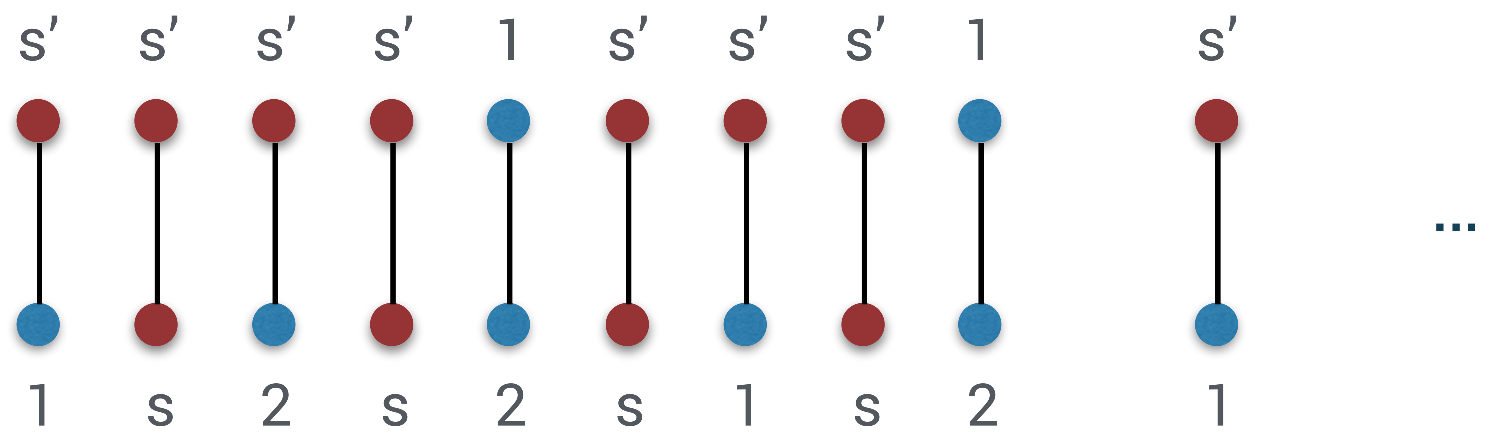
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



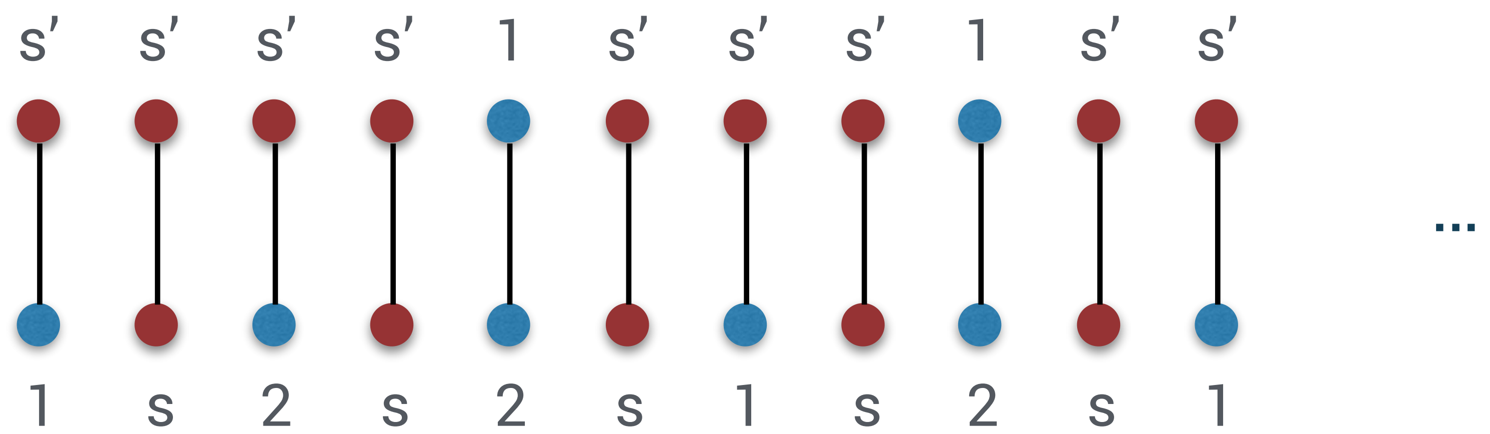
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



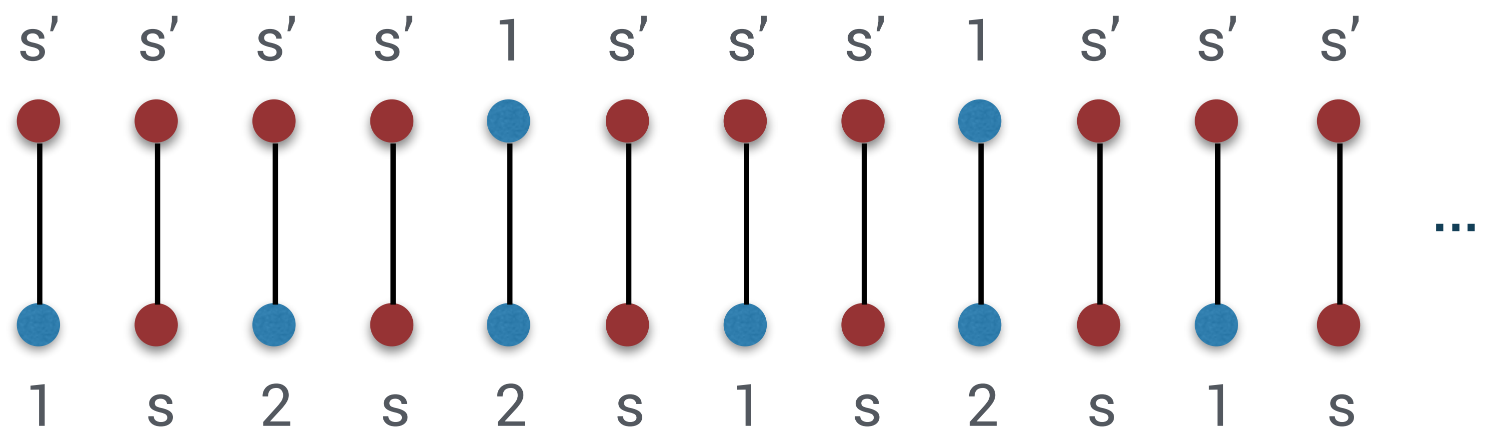
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



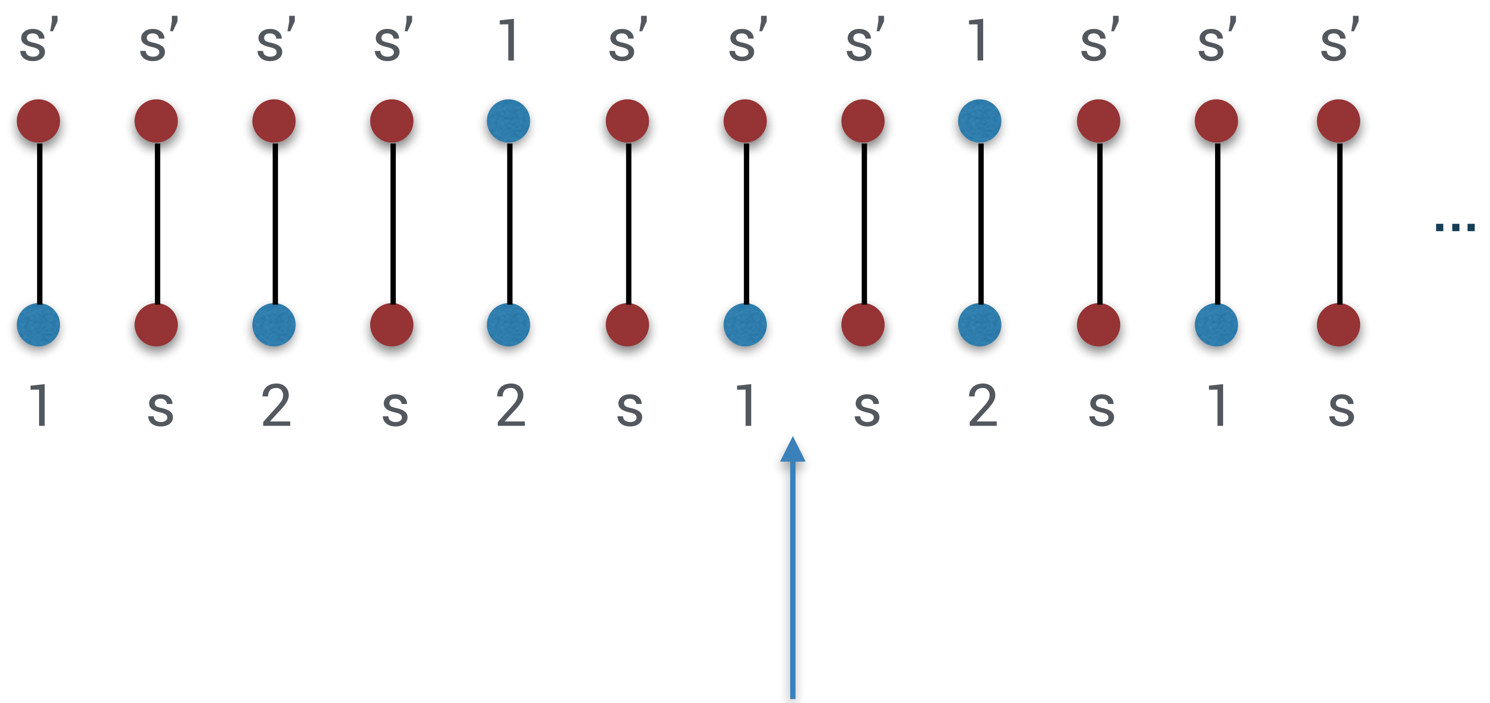
Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



Impossibility Results - Online Adaptive Adversary

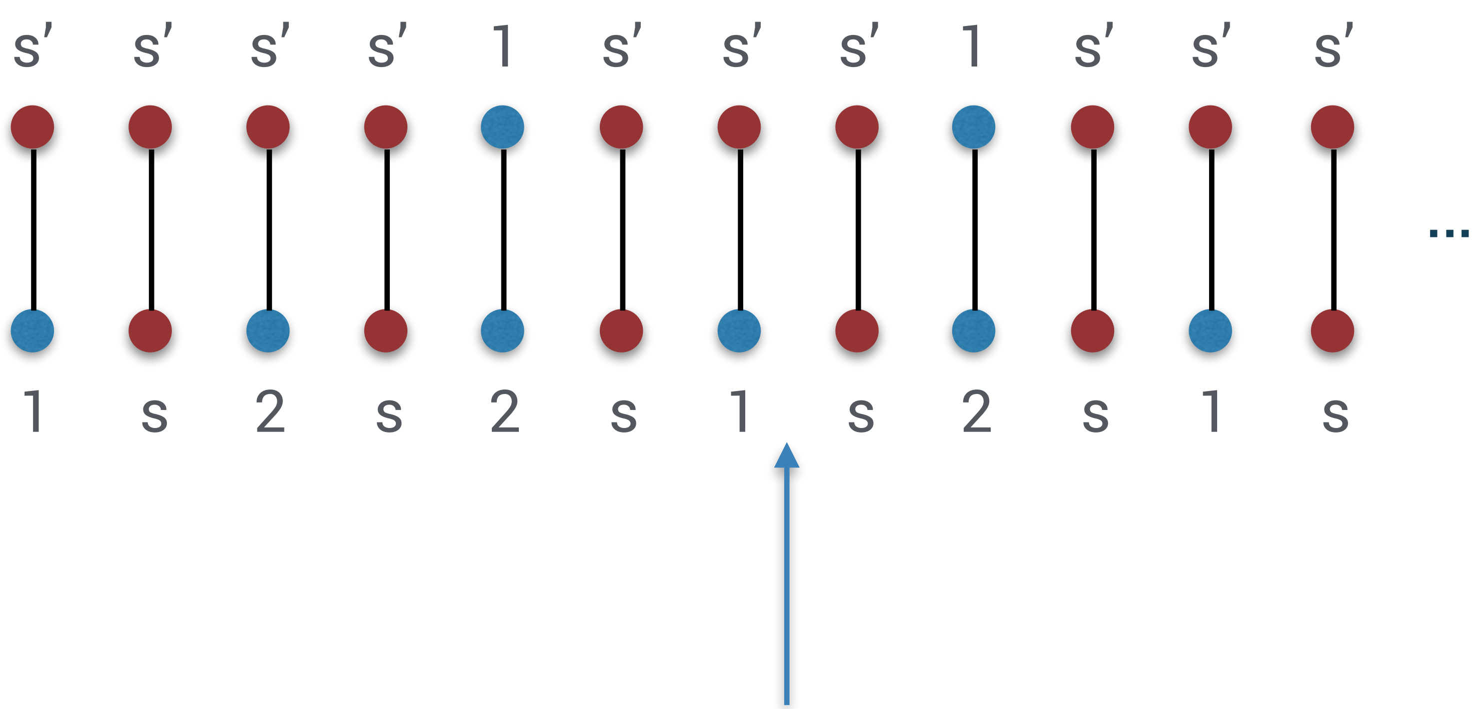
What if nodes are allowed to transmit more than once?



if s’ owns all the data:

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?

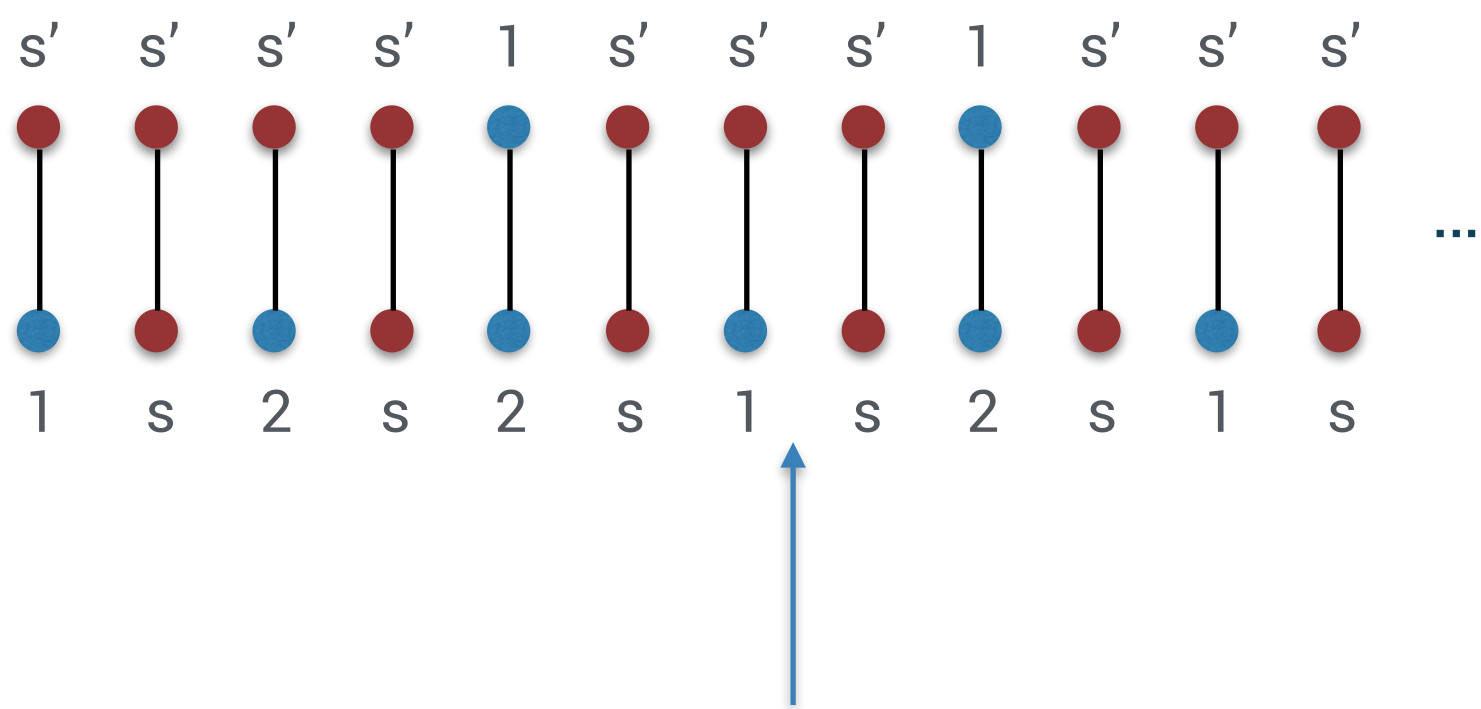


if s’ owns all the data:

then, a node has transmitted twice

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



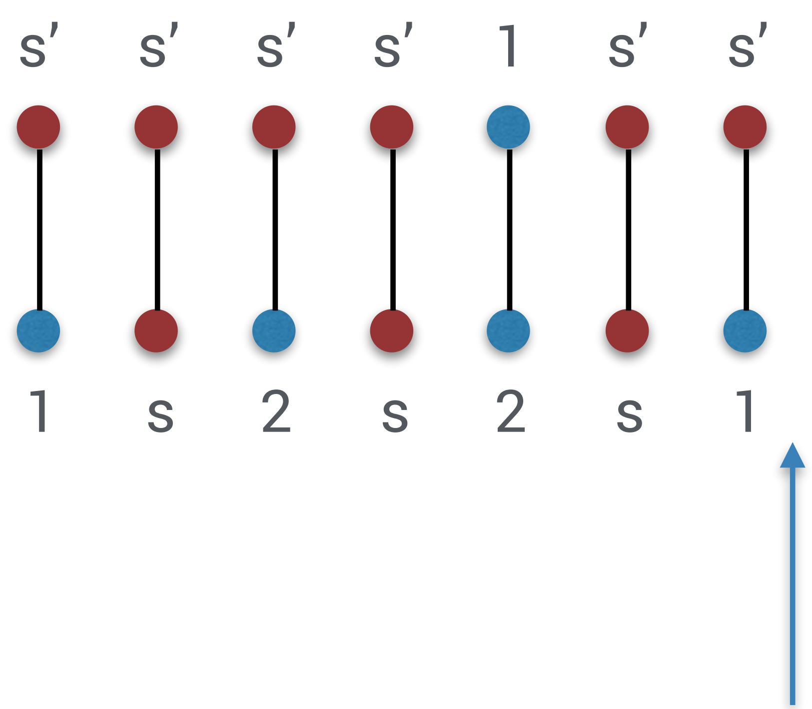
if s' owns all the data:

then, a node has transmitted twice

and s does not own all the data

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



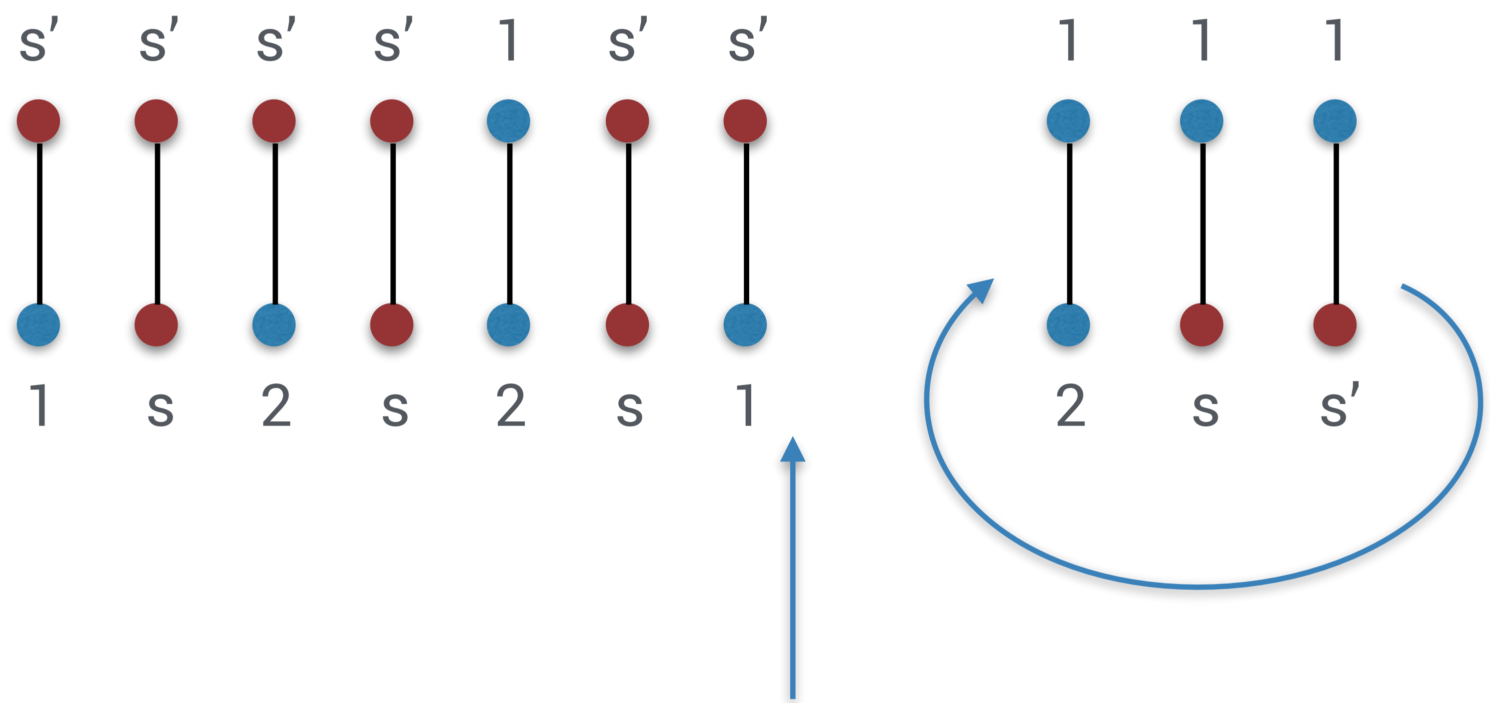
if s’ owns all the data:

then, a node has transmitted twice

and s does not own all the data

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



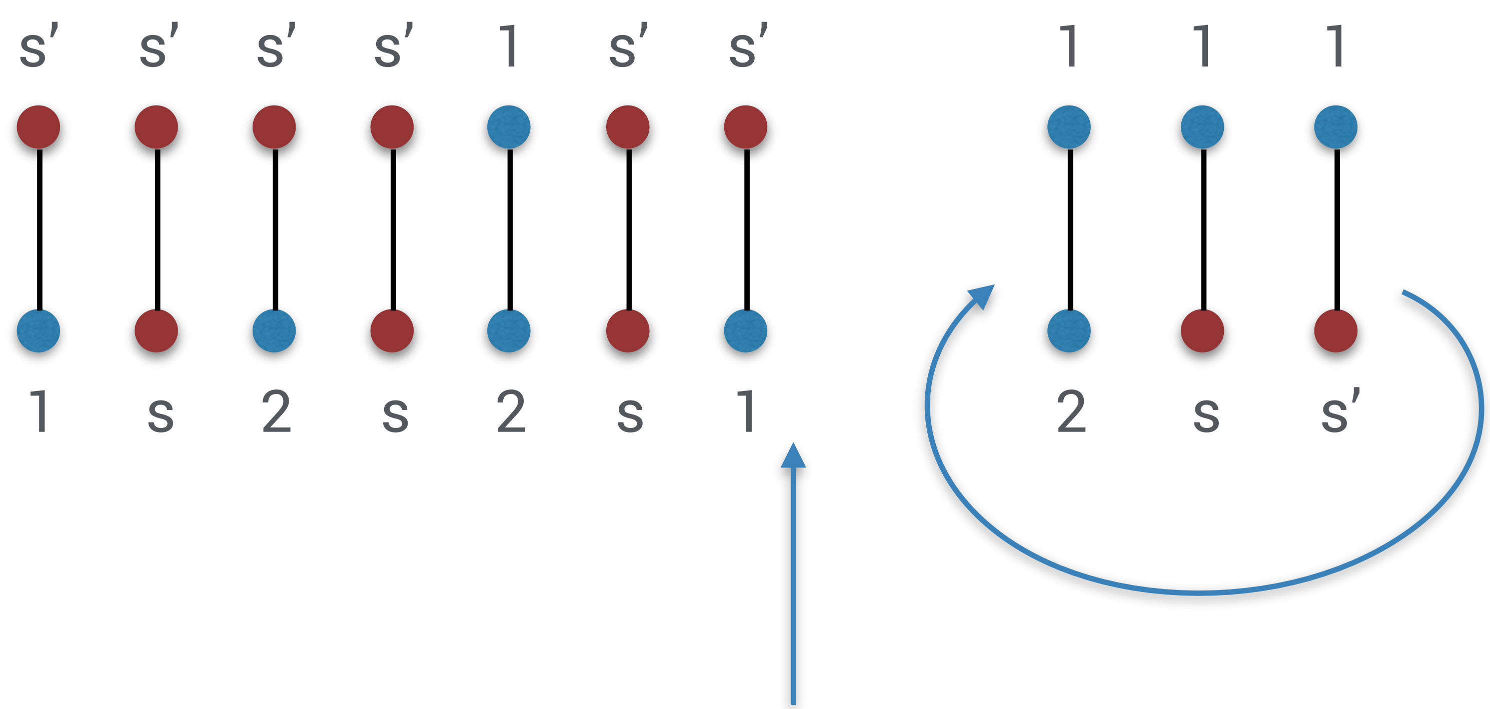
if s’ owns all the data:

then, a node has transmitted twice

and s does not own all the data

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



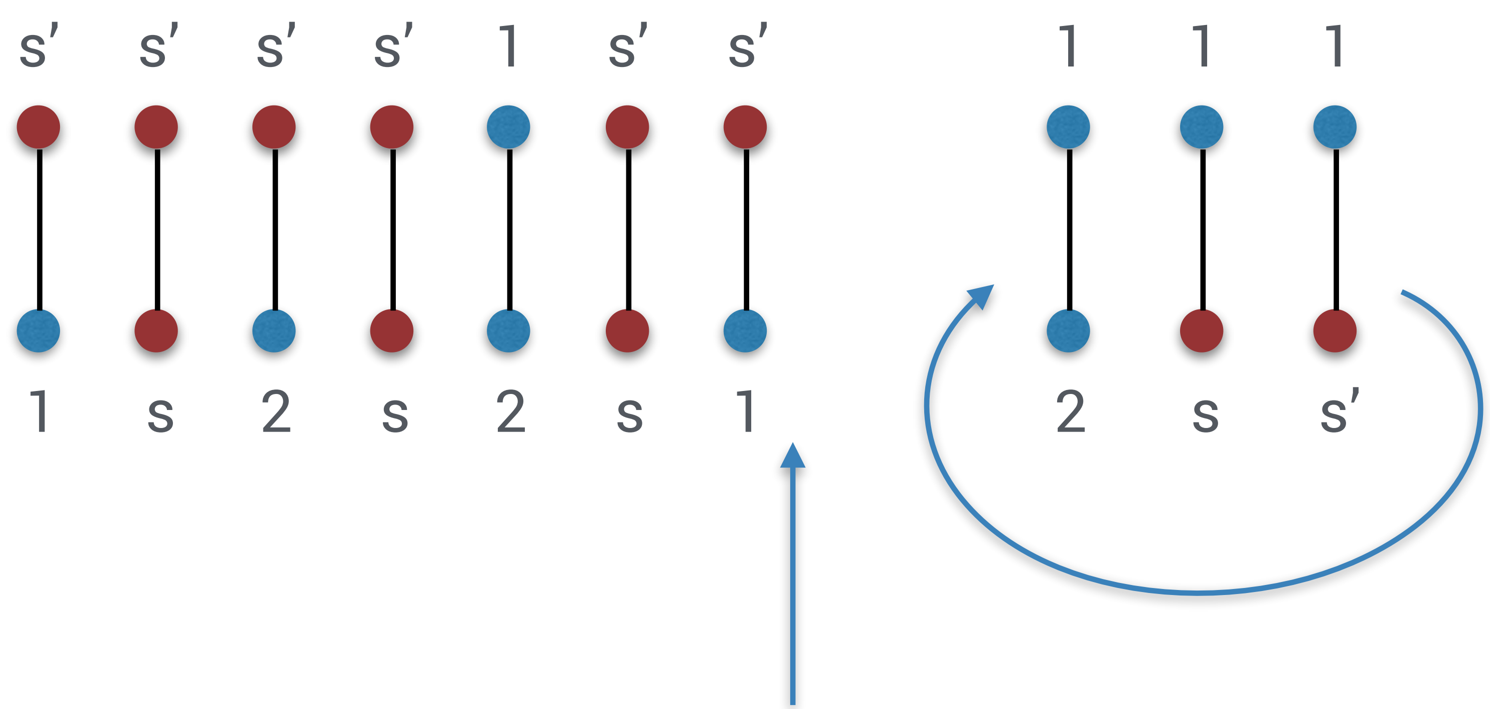
if s’ owns all the data:

then, a node has transmitted twice

and s will never own all the data

Impossibility Results - Online Adaptive Adversary

What if nodes are allowed to transmit more than once?



if s’ owns all the data:

then, a node has transmitted twice

and s will never own all the data

yet a convergecast is always possible