Benchmarking API Costs of Network Sampling Strategies

Michele Coscia & Luca Rossi
ITU København

August 27th, 2019

IT UNIVERSITY OF COPENHAGEN

(i) Why Network Sampling?

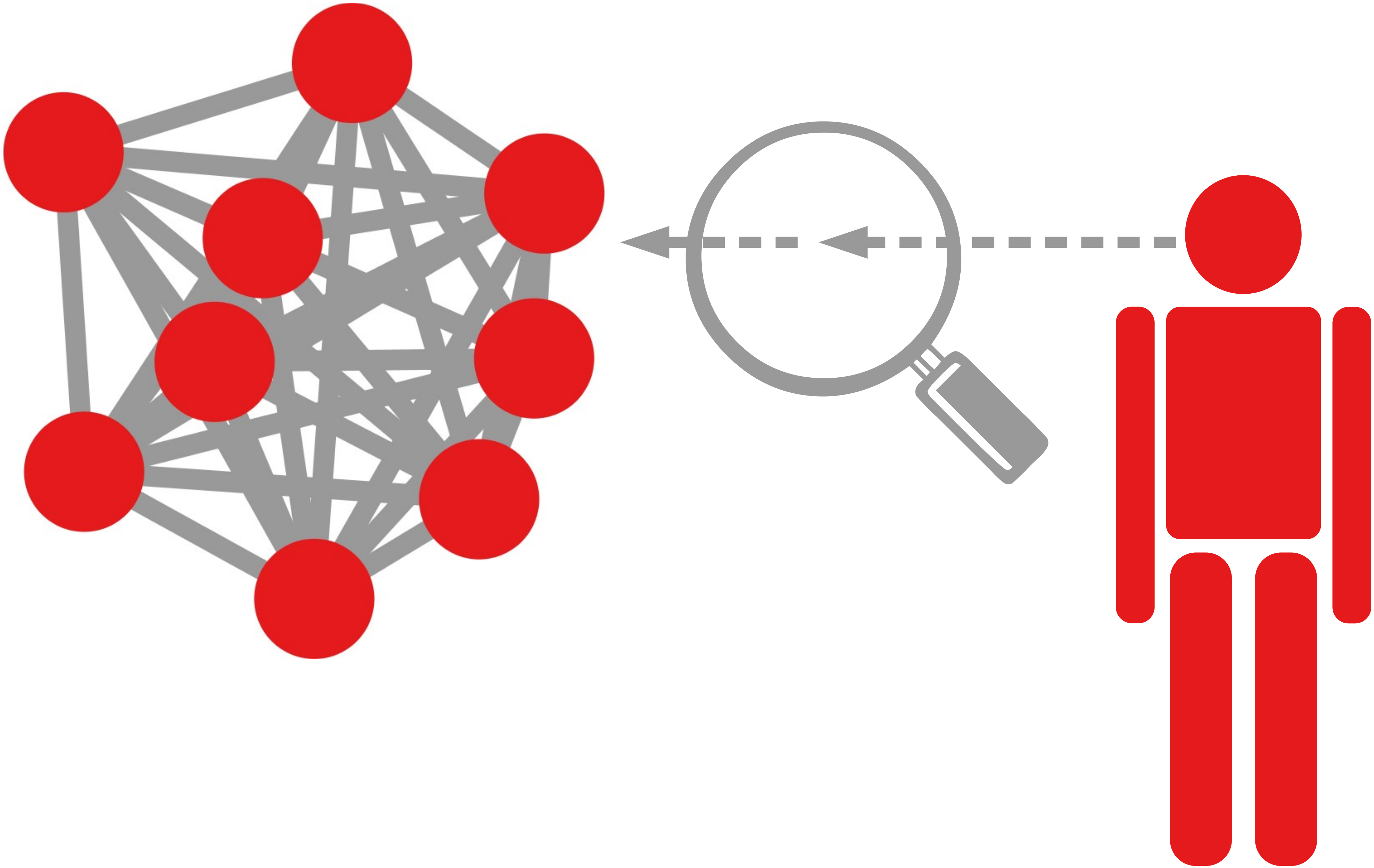
(i) Why Network Sampling?

(ii) Are there understudied real world obstacles that should make us reconsider how we choose the best sampling strategy?

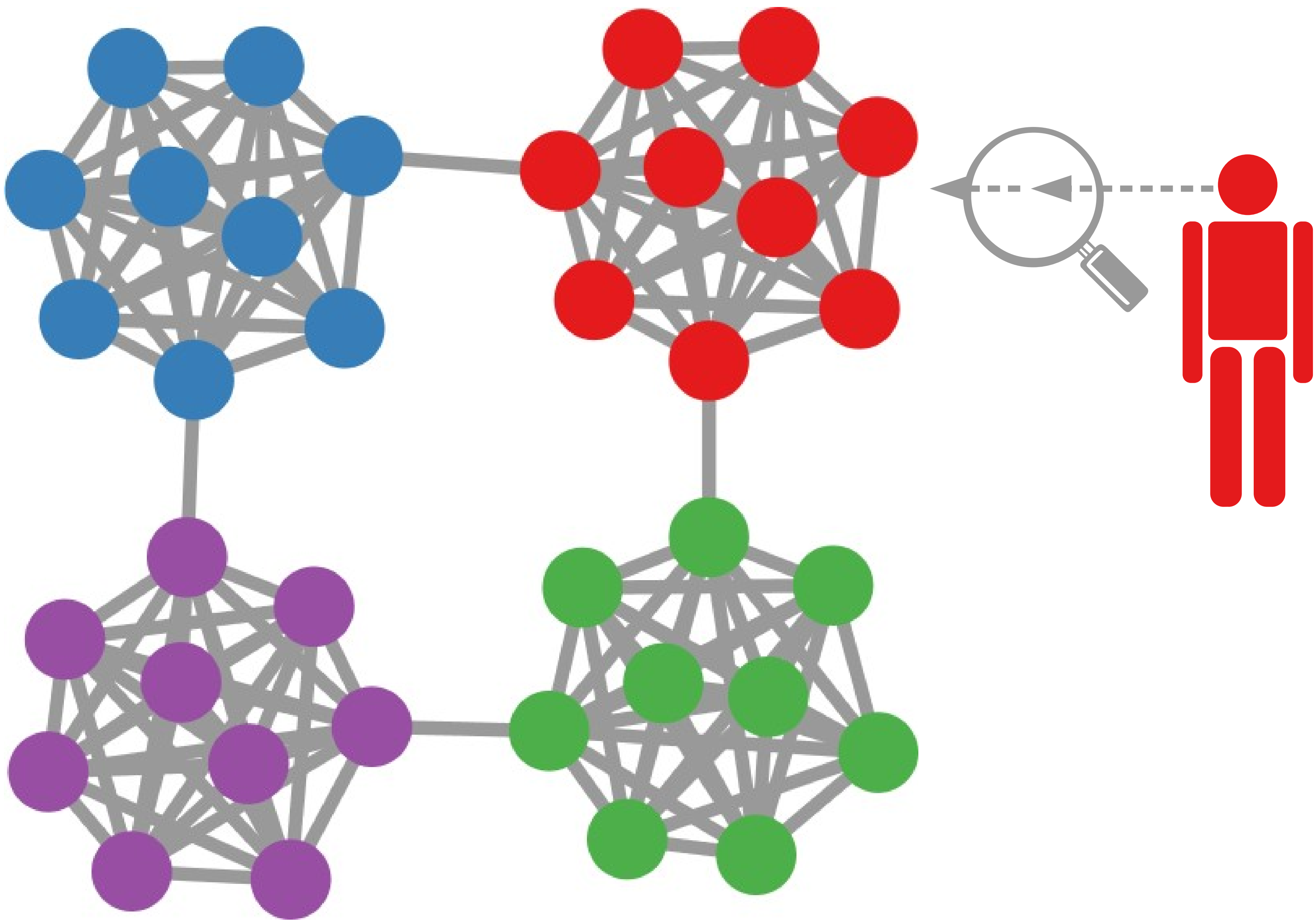
(i) Why Network Sampling?

The Observation Problem

The Observation Problem



The Observation Problem



~330M monthly users

~330M monthly users

~1.1m per user

~330M monthly users

~1.1m per user

~21.78B seconds

~330M monthly users

~1.1m per user

~21.78B seconds

~690 years

~330M monthly users

~1.1m per user

~21.78B seconds

~690 years

1 year of crawling

Network Exploration

- BFS, DFS

- Random Walks
- Snowball
- Forest Fire

Network Exploration

- BFS, DFS ⟼ **Full exploration as the objective**
- Random Walks
- Snowball
- Forest Fire

Network Exploration

- BFS, DFS

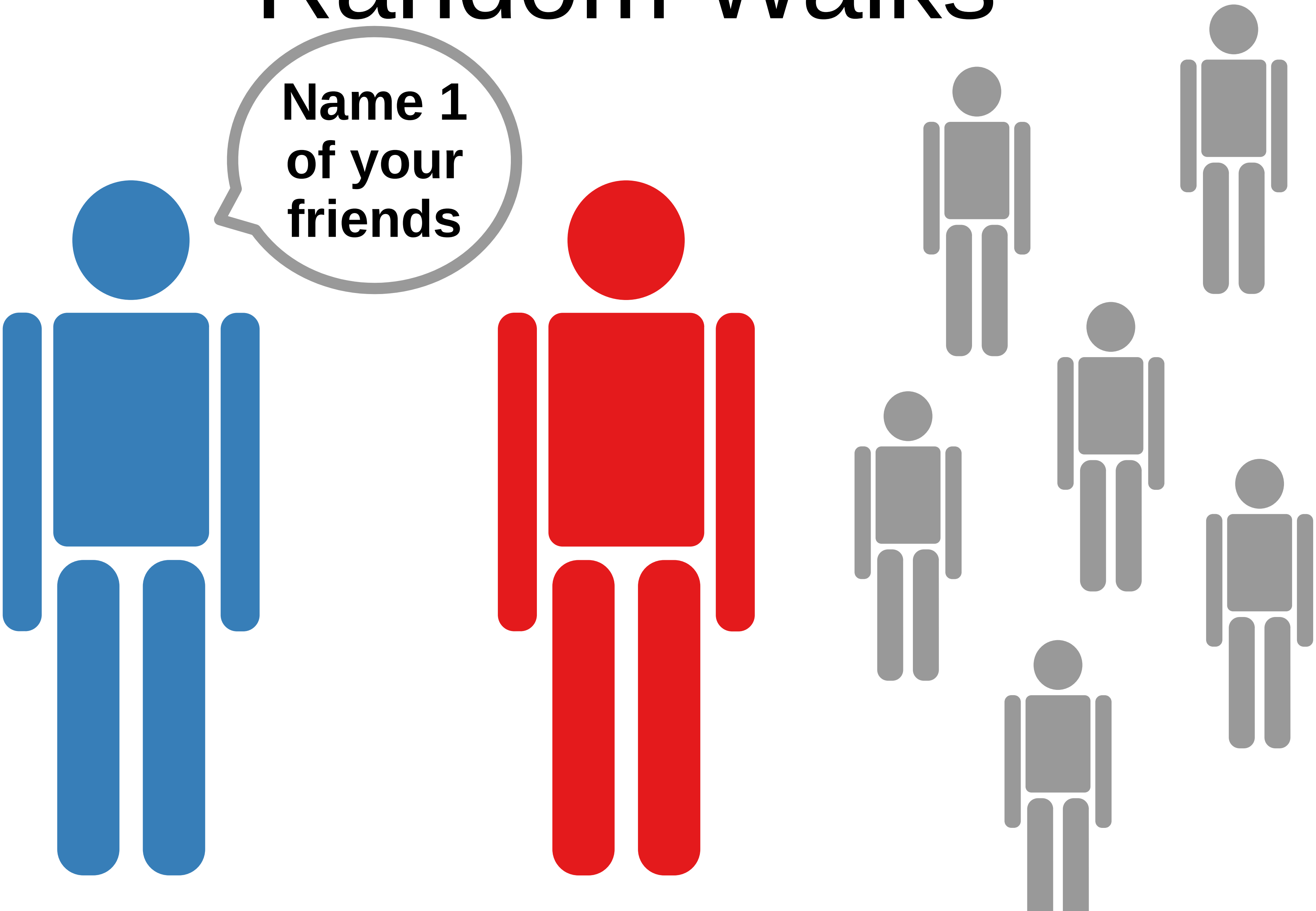
Full exploration as the objective

- Random Walks
- Snowball
- Forest Fire

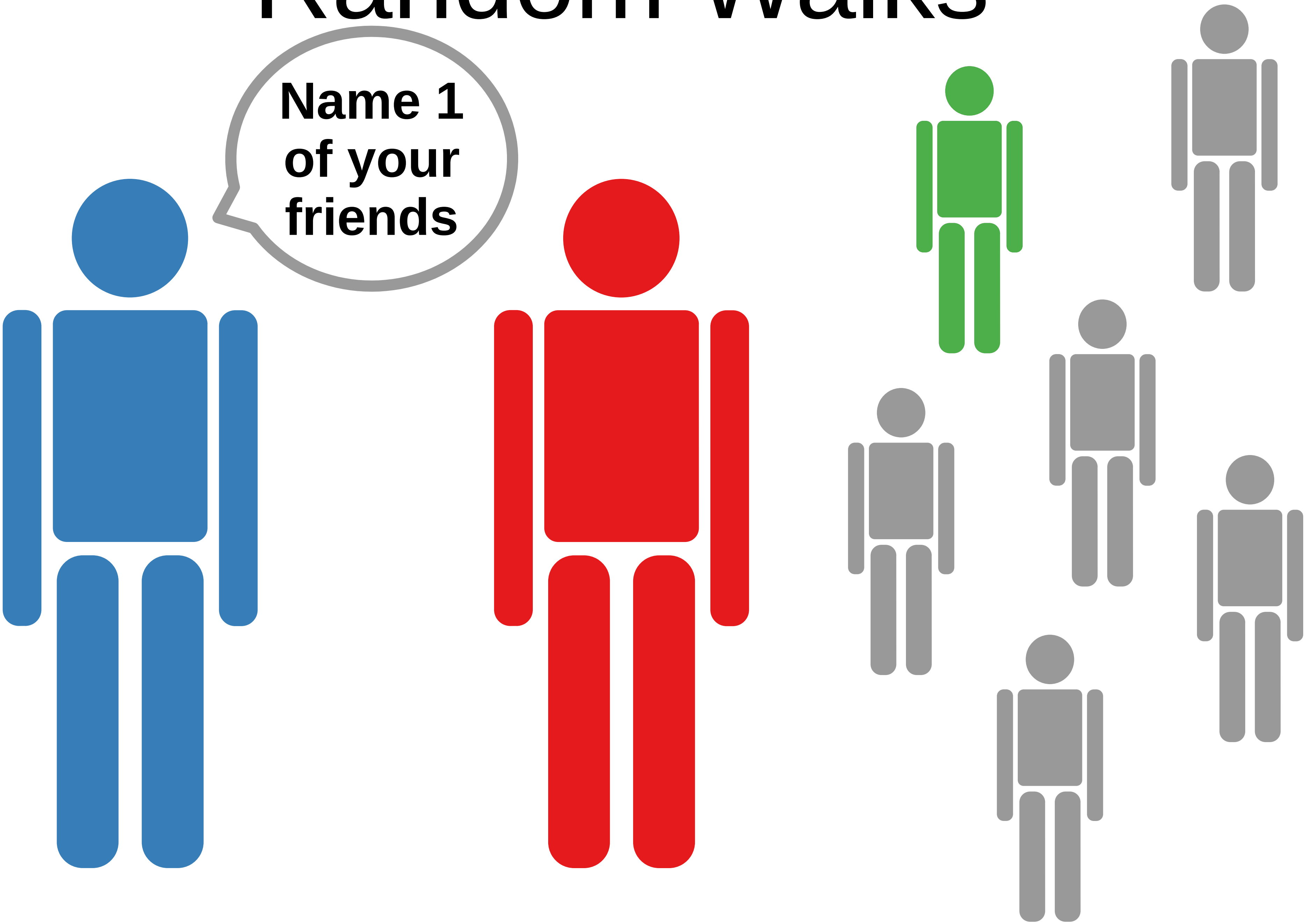
Preventing bias from samples

Random Walks

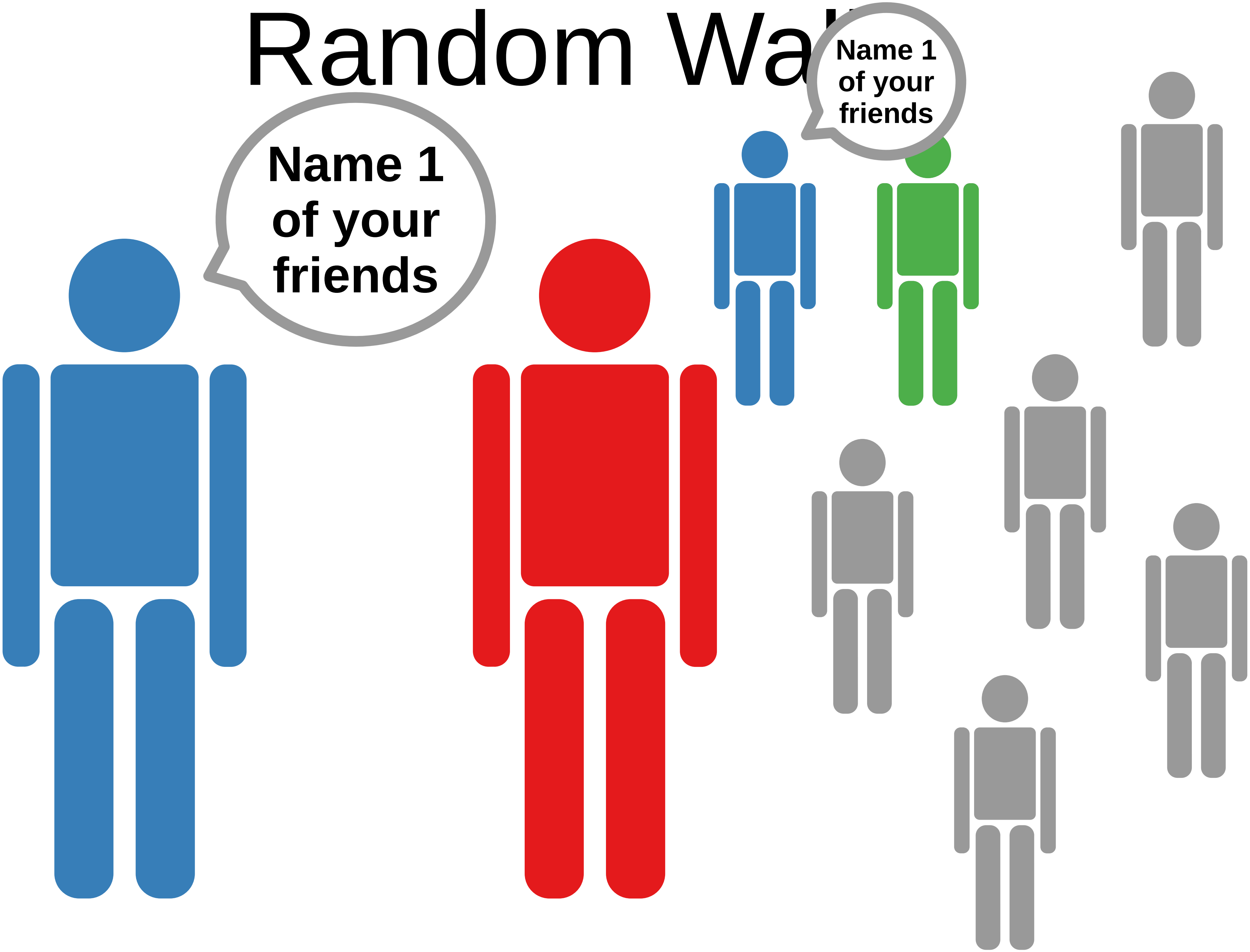
Random Walks



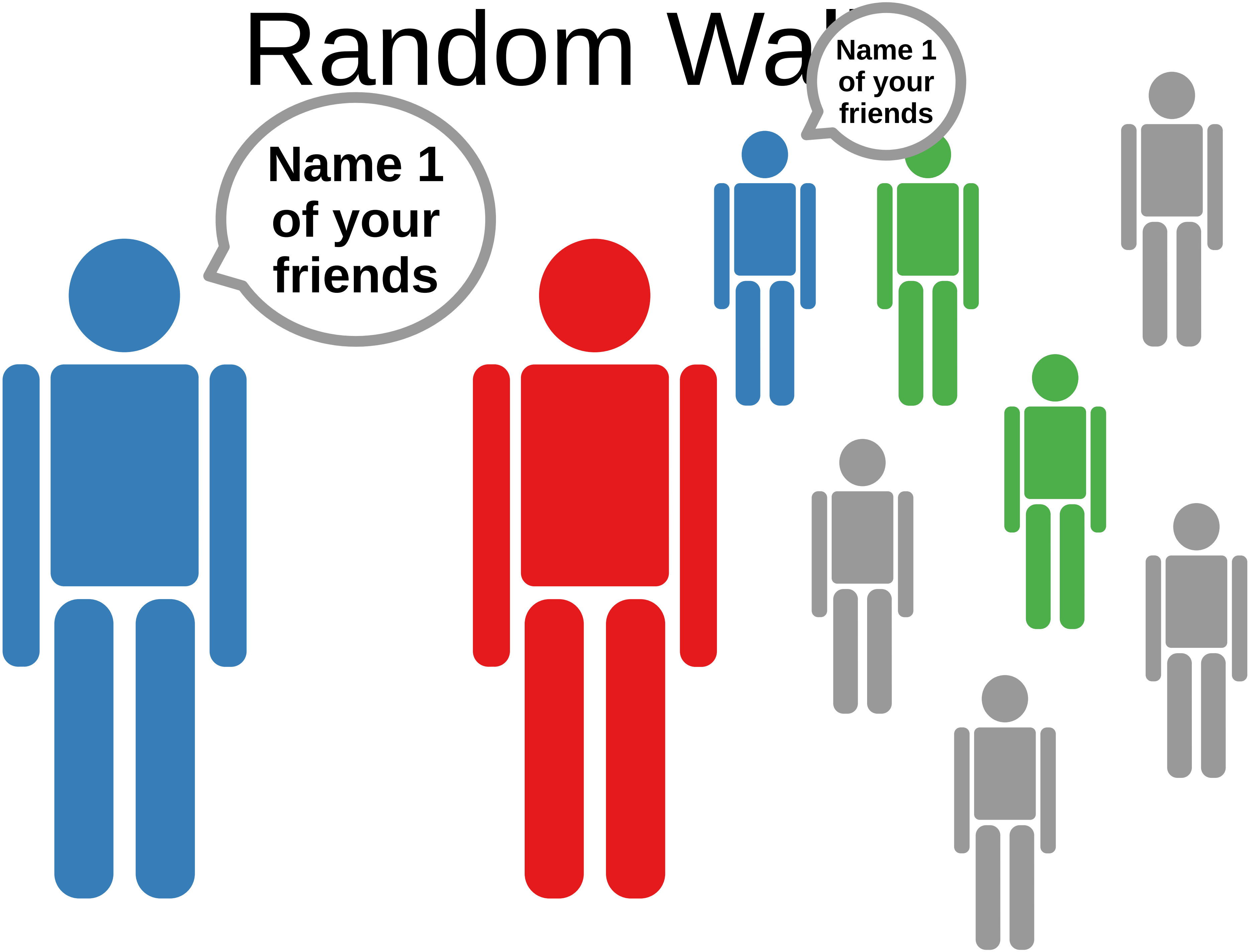
Random Walks



Random Wa



Random Wal



Degree Bias

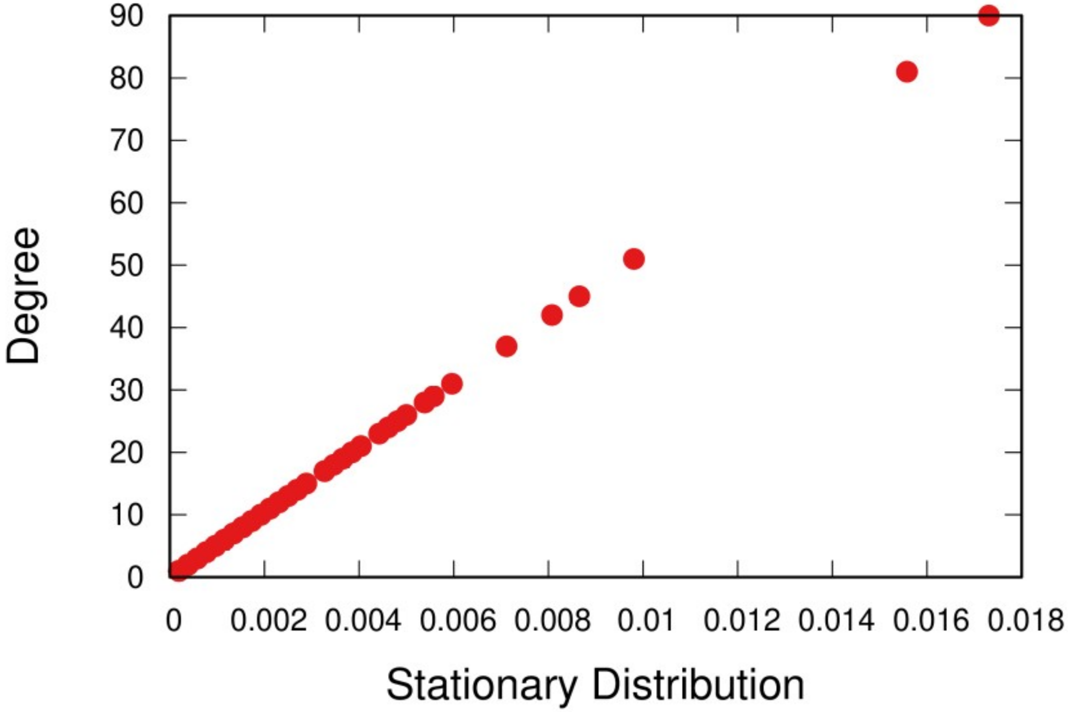
Degree Bias

- Stationary distr
  π

0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2
0.09	0.1	0.09	0.06	0.09	0.2	0.06	0.1	0.2

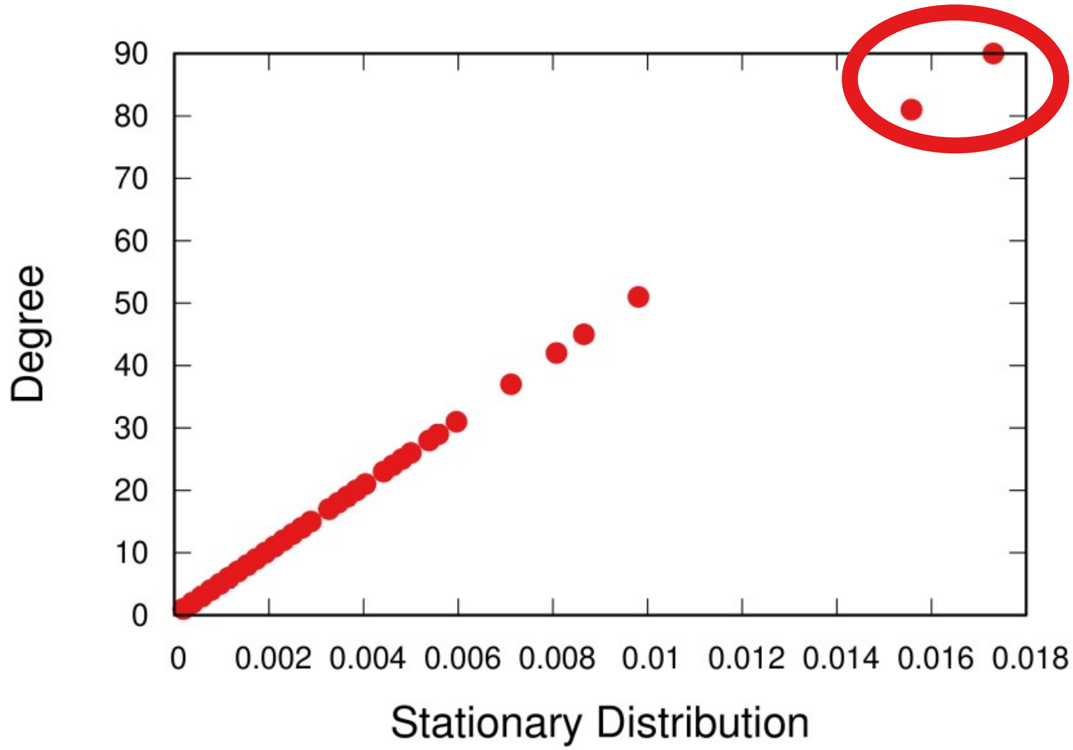
Degree Bias

- Stationary distr π
- π = degree

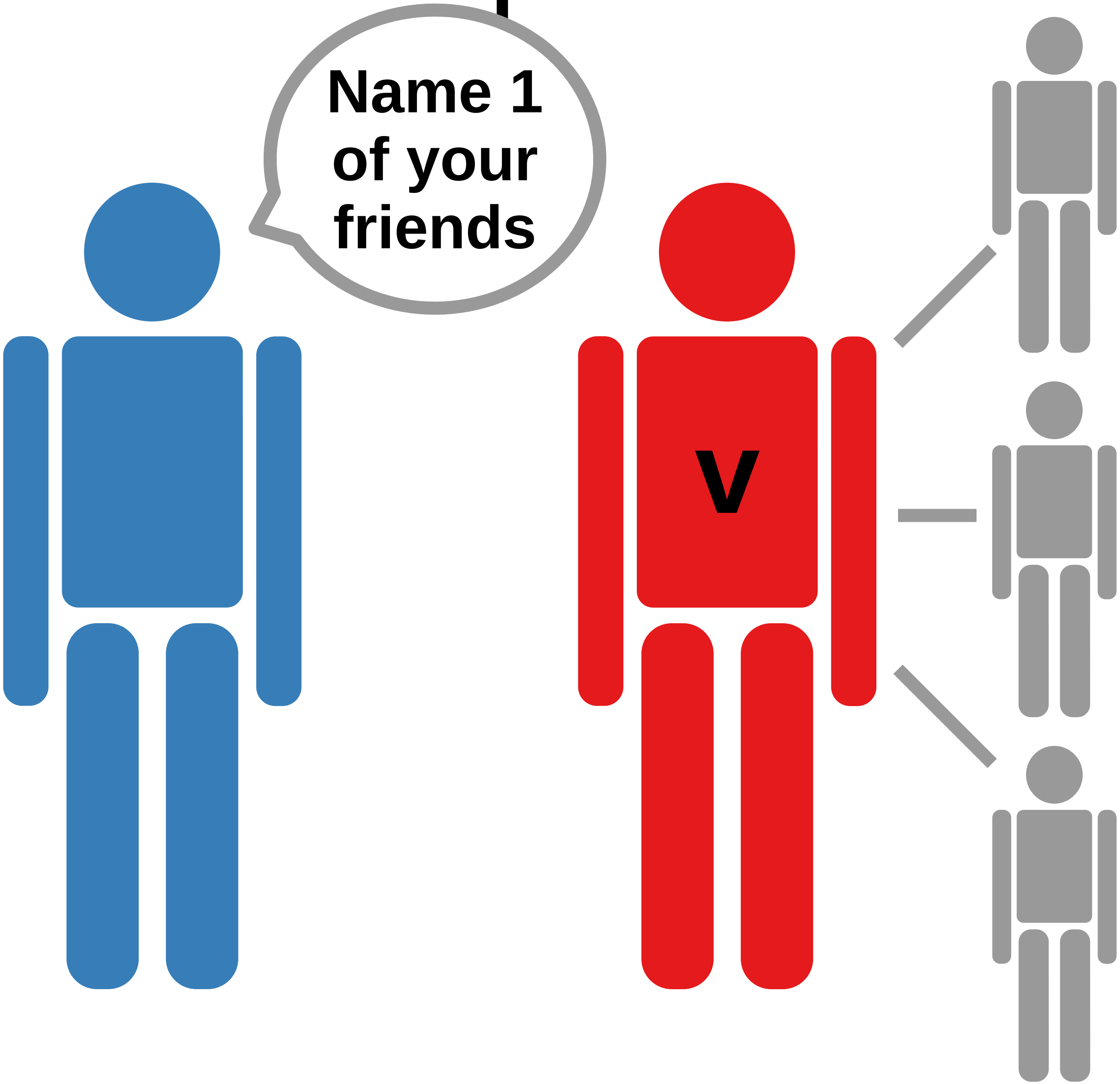


Degree Bias

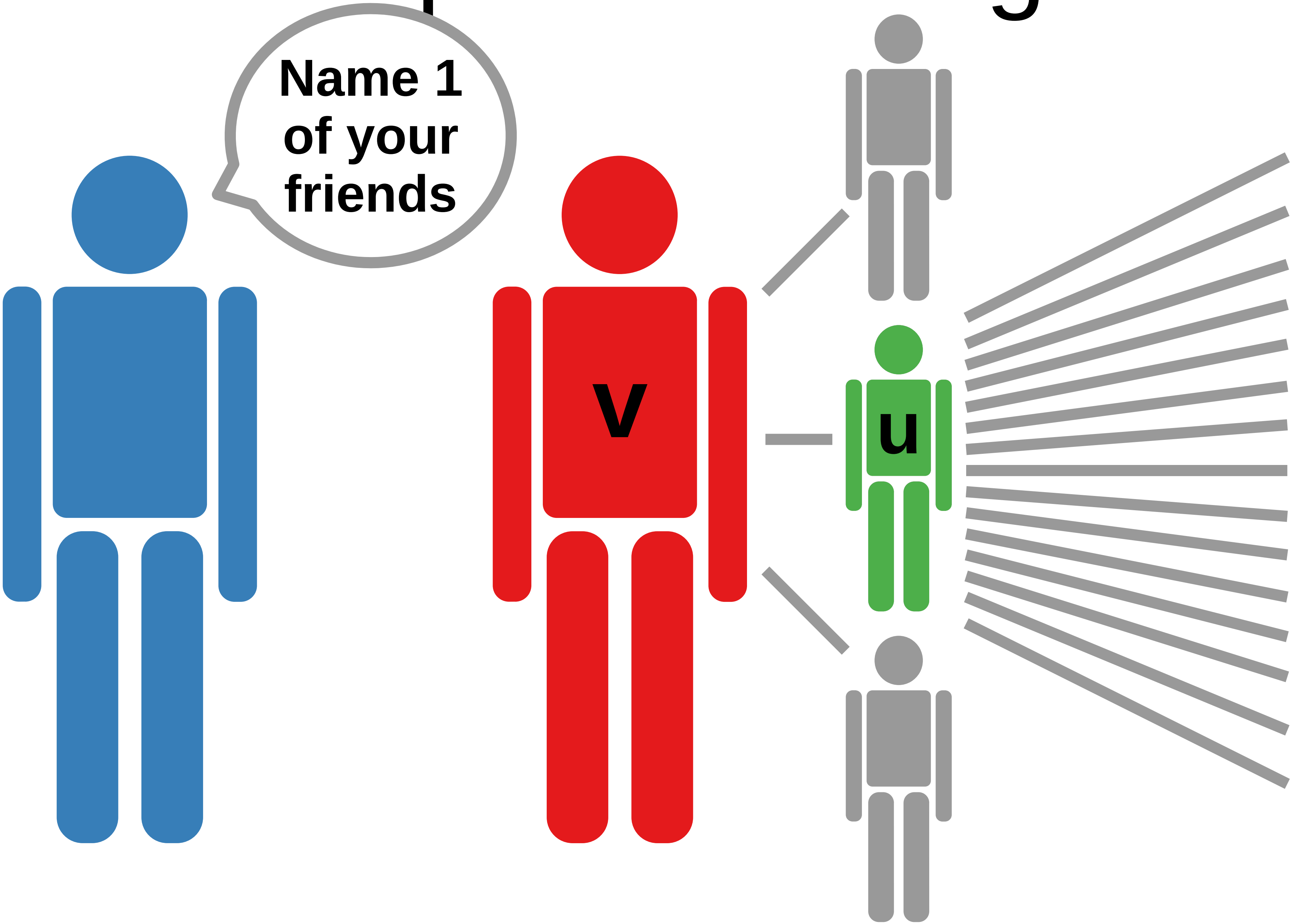
- Stationary distr π
- π = degree
- Oversampled hubs!



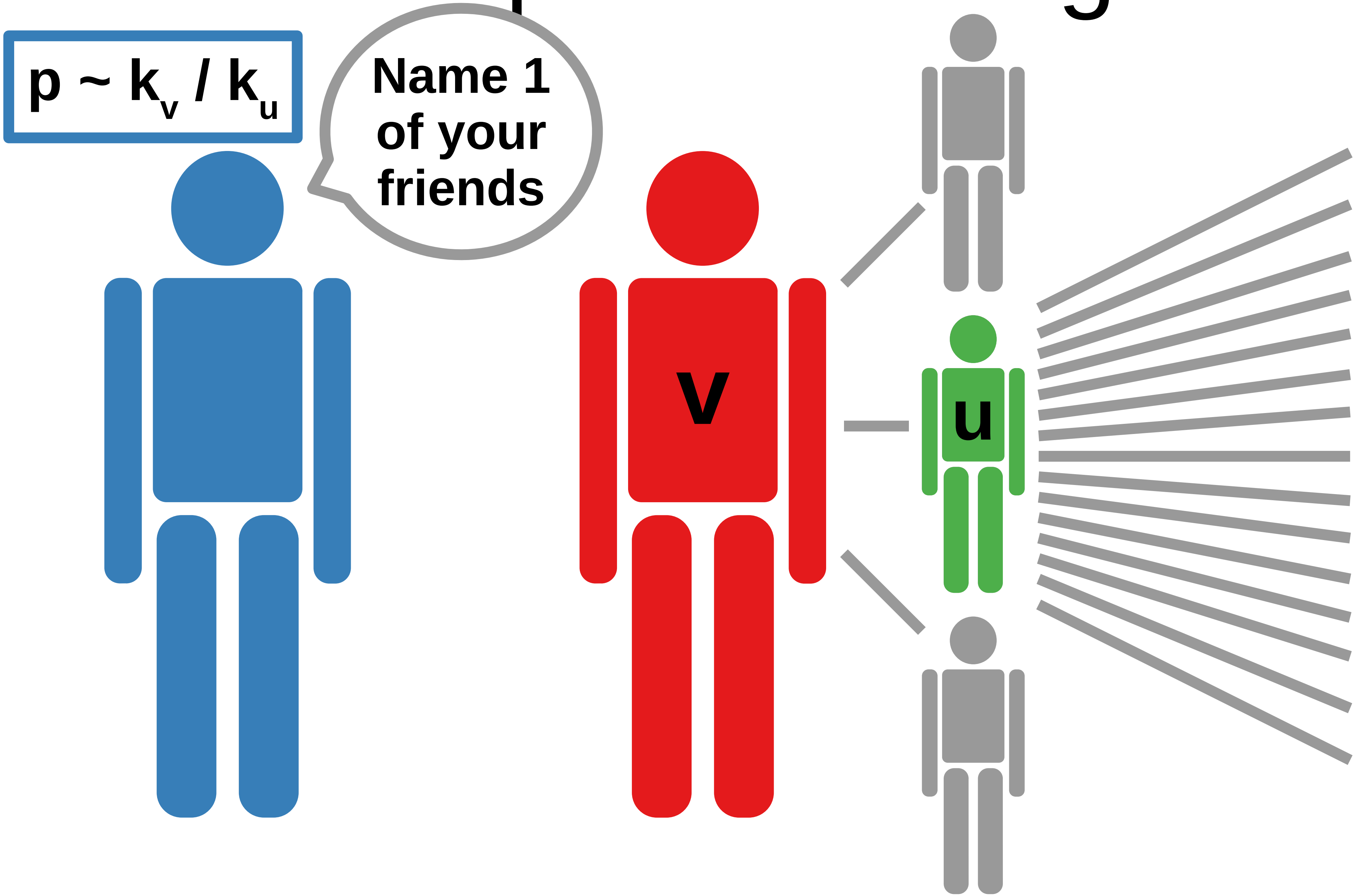
Metropolis-Hastings



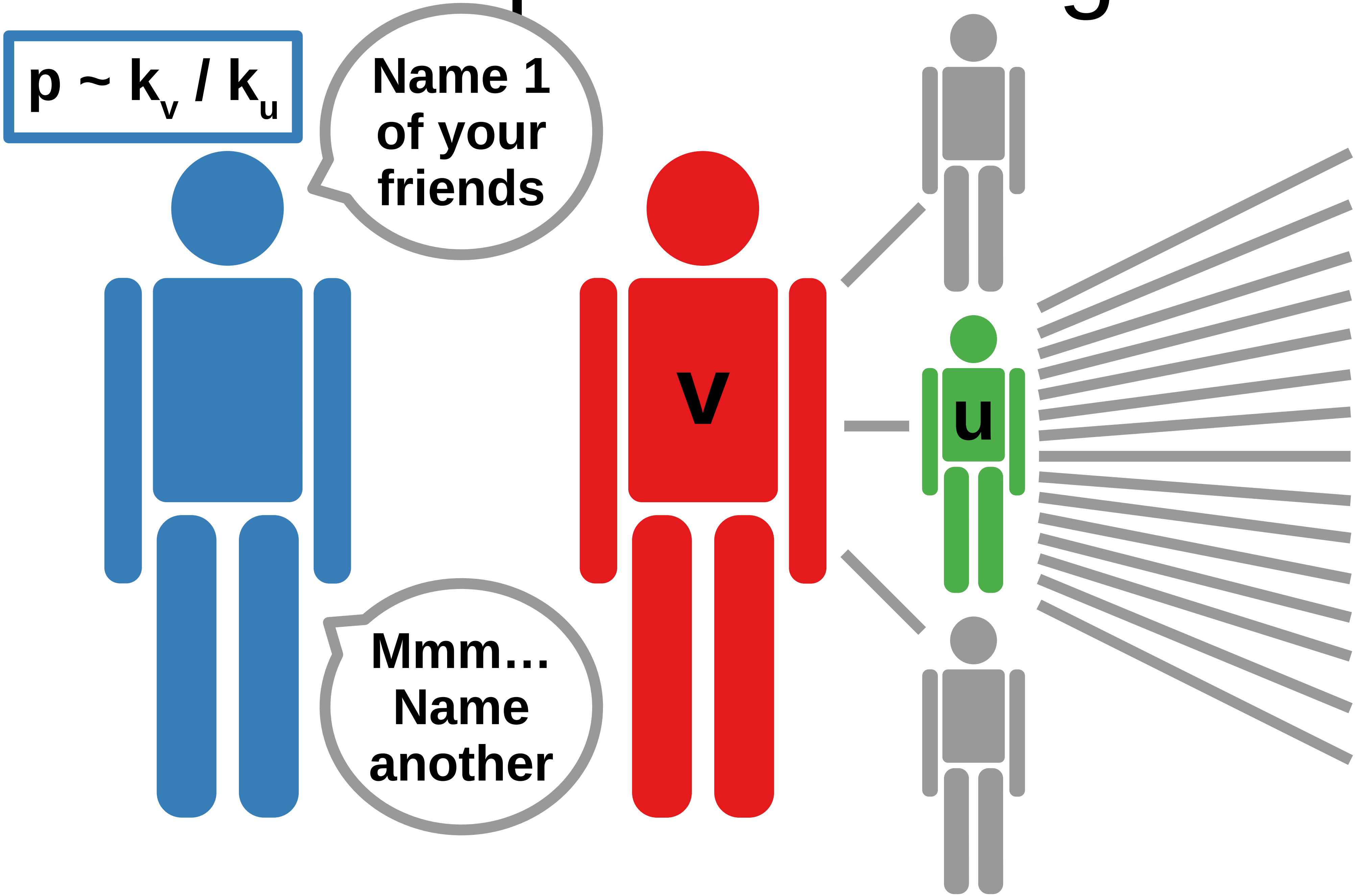
Metropolis-Hastings



Metropolis-Hastings



Metropolis-Hastings



Re-Weighted RW

Re-Weighted RW

- Perform vanilla RW

Re-Weighted RW

- Perform vanilla RW

- Re-weight property of interest

Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest

$$p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} x_{v'}^{-1}}$$

Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest

p of nodes with value i

$$\boxed{p_i} = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} x_{v'}^{-1}}$$

Re-Weighted RW

- Perform vanilla RW

- Re-weight property of interest

p of nodes with value i

$$\boxed{p_i} = \frac{\sum_{v \in \boxed{V_i}} i^{-1}}{\sum_{v' \in V} x_{v'}^{-1}}$$

Set of nodes with value i

Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest

p of nodes with value i

$$\boxed{p_i} = \frac{\sum_{v \in \boxed{V_i}} i^{-1}}{\sum_{v' \in \boxed{V}} x_{v'}^{-1}}$$

Set of nodes with value i

Set of nodes in the sample

Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest

p of nodes with value i

$$\boxed{p_i} = \frac{\sum_{v \in \boxed{V_i}} i^{-1}}{\sum_{v' \in \boxed{V}} \boxed{x_{v'}}^{-1}}$$

Set of nodes with value i

Set of nodes in the sample

Value for v’

Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest
- Respondent-Driven Sampling

p of nodes
with value i

$$\boxed{p_i} = \frac{\sum_{v \in \boxed{V_i}} i^{-1}}{\sum_{v' \in \boxed{V}} \boxed{x_{v'}}^{-1}}$$

Set of
nodes
with
value i

Value
for v’

Set of nodes
in the sample

Re-Weighted RW: Example

Re-Weighted RW: Example

- p of a node
  having k=2?

Re-Weighted RW: Example

- p of a node having k=2?
- Observed: 20 over 100 (p = 0.2)

Re-Weighted RW: Example

- p of a node having k=2?
- Observed: 20 over 100 (p = 0.2)
- Other nodes:
  - k=1: 50
  - k=3: 10
  - k=4: 8
  - k=5: 7
  - k=6: 5

Re-Weighted RW: Example

- p of a node having k=2?
- Observed: 20 over 100 (p = 0.2)
- Other nodes:
  - k=1: 50
  - k=3: 10
  - k=4: 8
  - k=5: 7
  - k=6: 5

$$p_2 = \frac{20 * 1/2}{(50/1)+(20/2)+(10/3)+(8/4)+(7/5)+(5/6)}$$

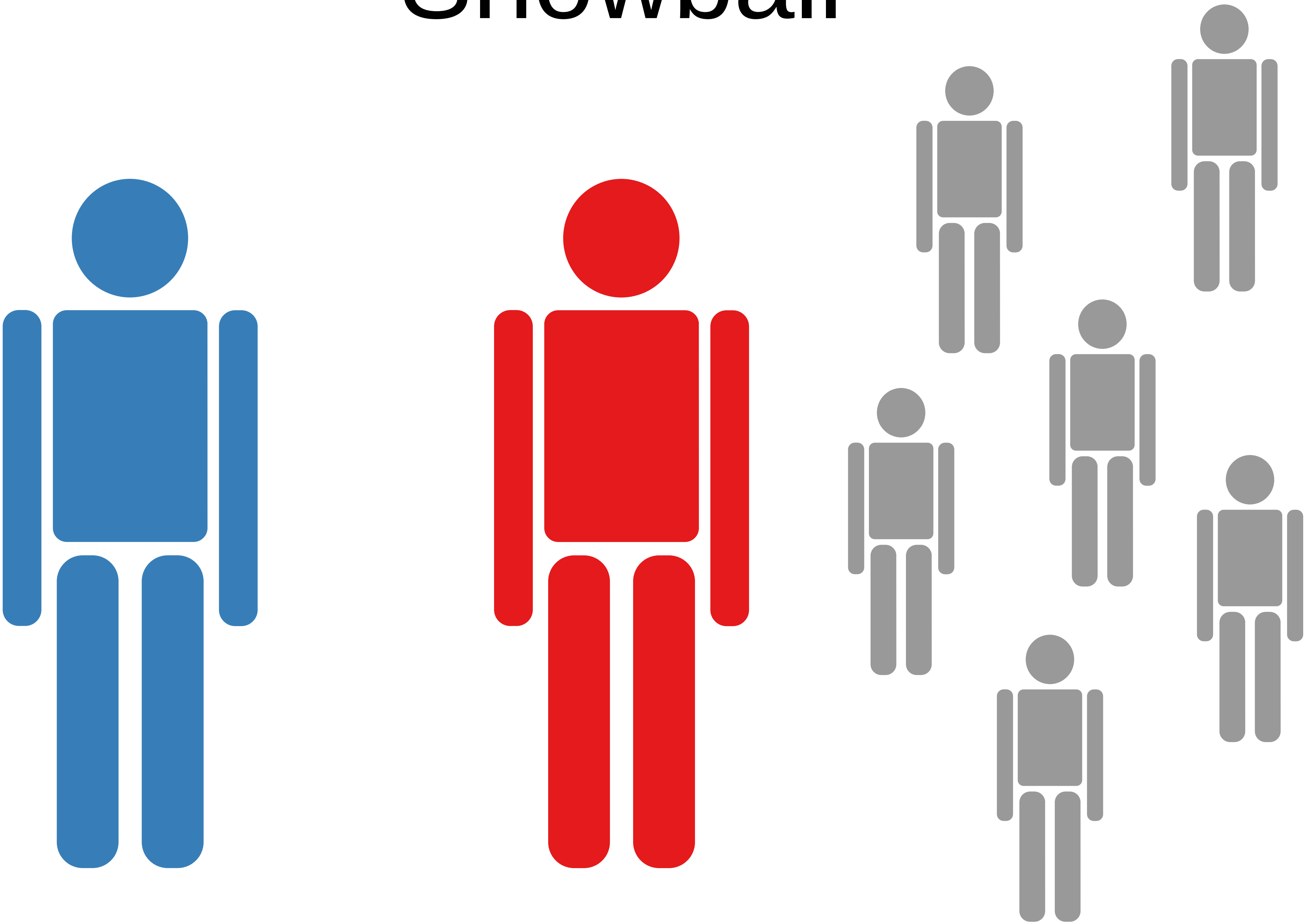
Re-Weighted RW: Example

- p of a node having k=2?
- Observed: 20 over 100 (p = 0.2)
- Other nodes:
  - k=1: 50
  - k=3: 10
  - k=4: 8
  - k=5: 7
  - k=6: 5

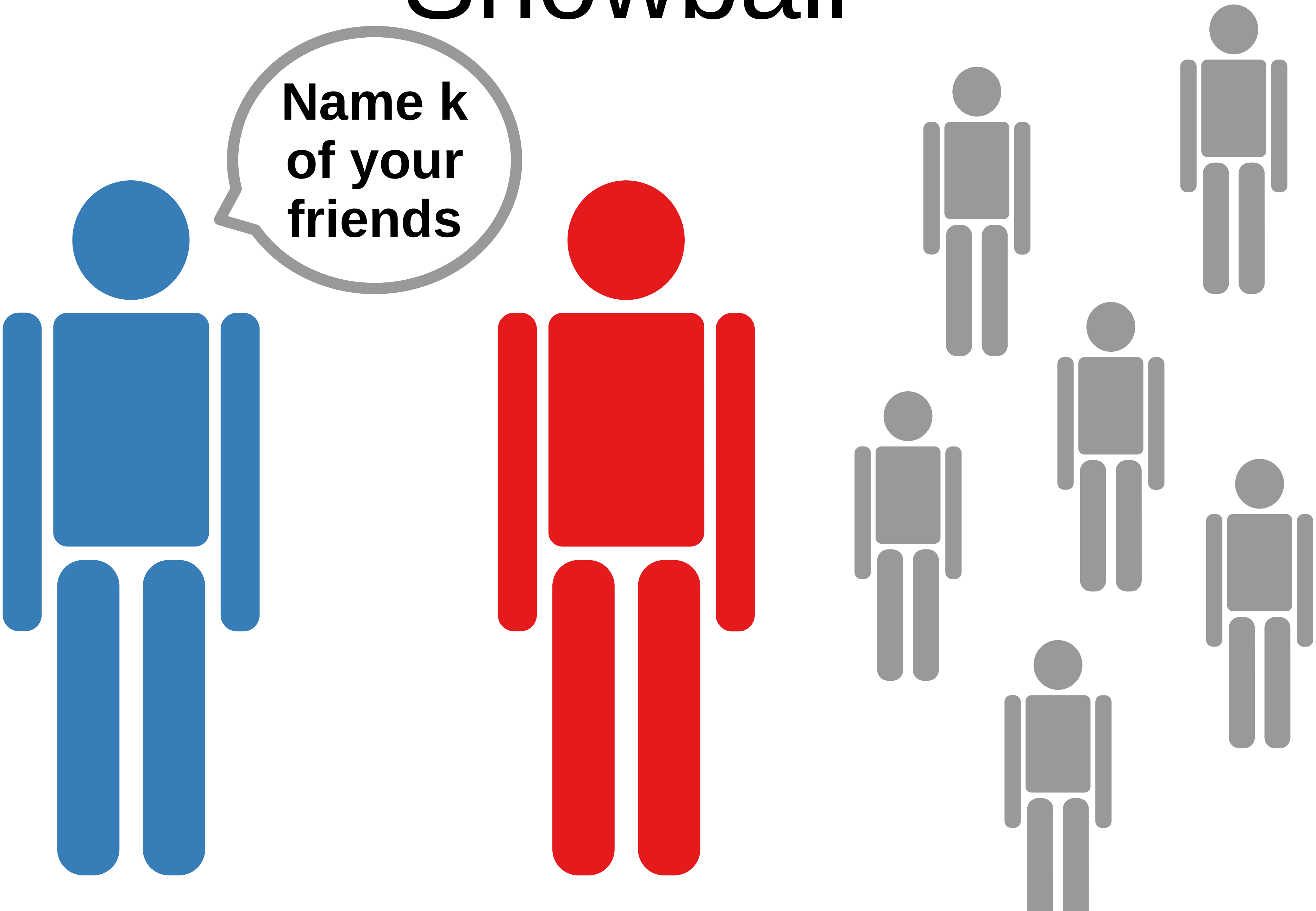
$$p_2 = \frac{20 * 1/2}{(50/1)+(20/2)+(10/3)+(8/4)+(7/5)+(5/6)}$$

$$p_2 = \frac{10}{67.5\overline{6}} \sim 0.148$$

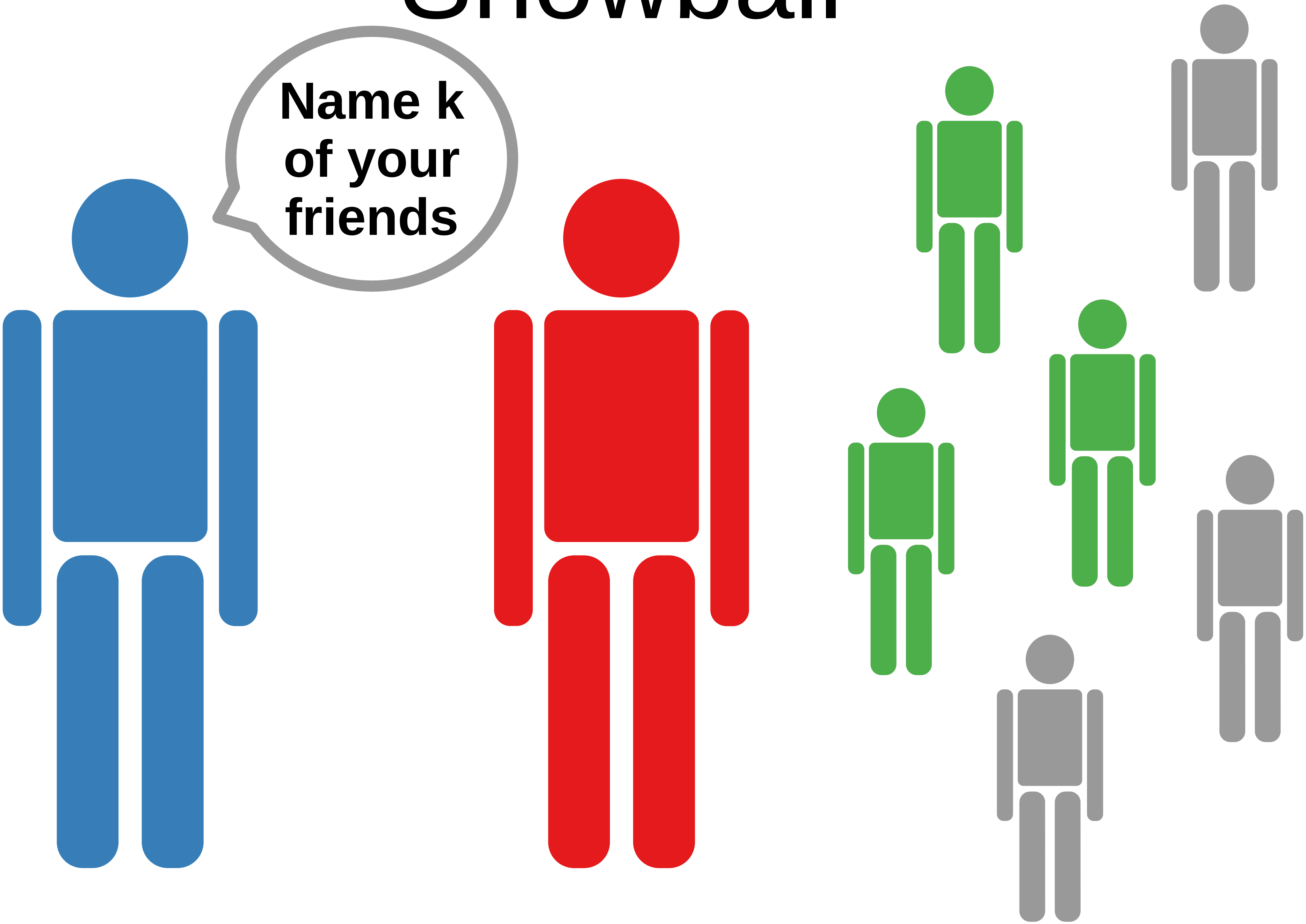
Snowball



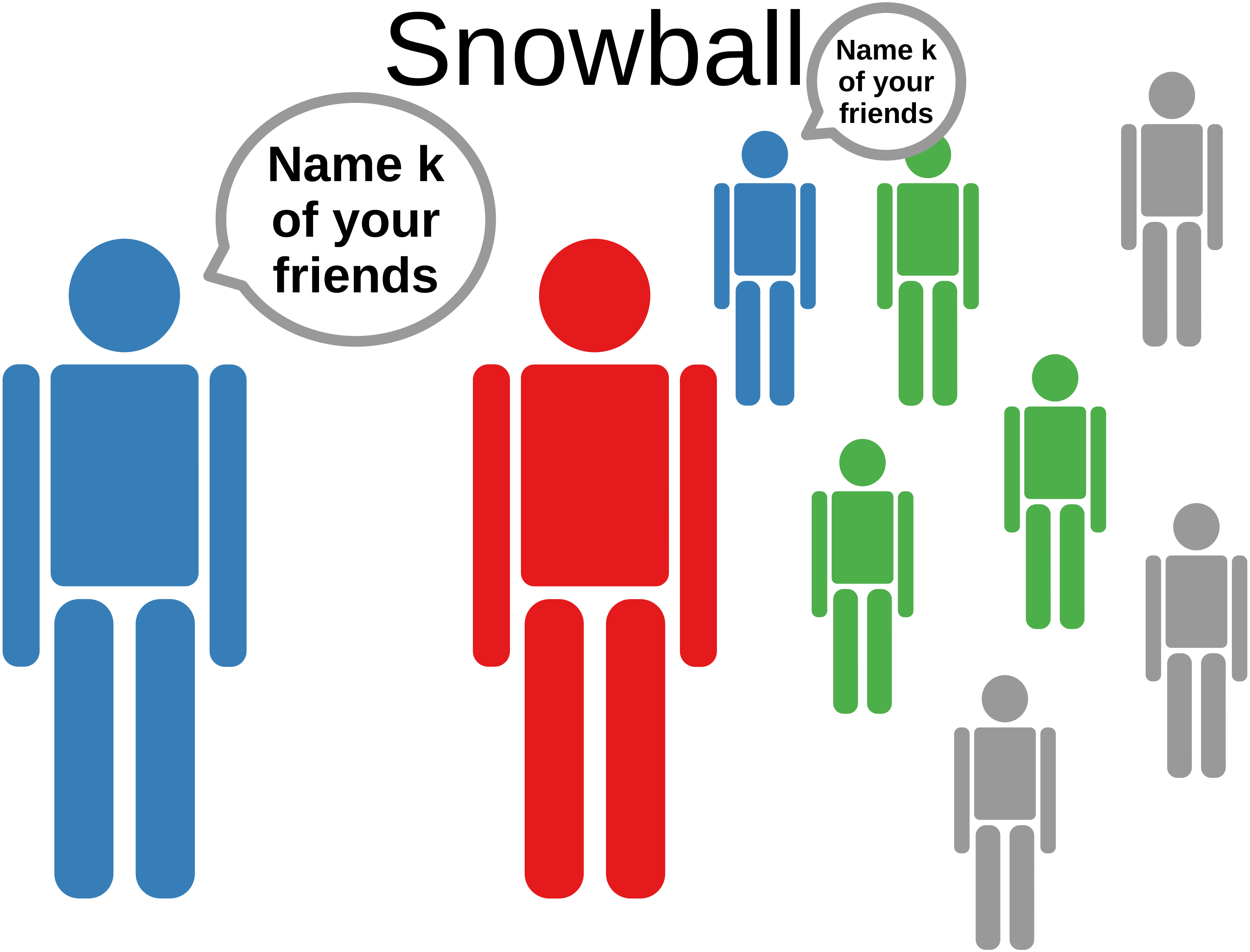
Snowball



Snowball



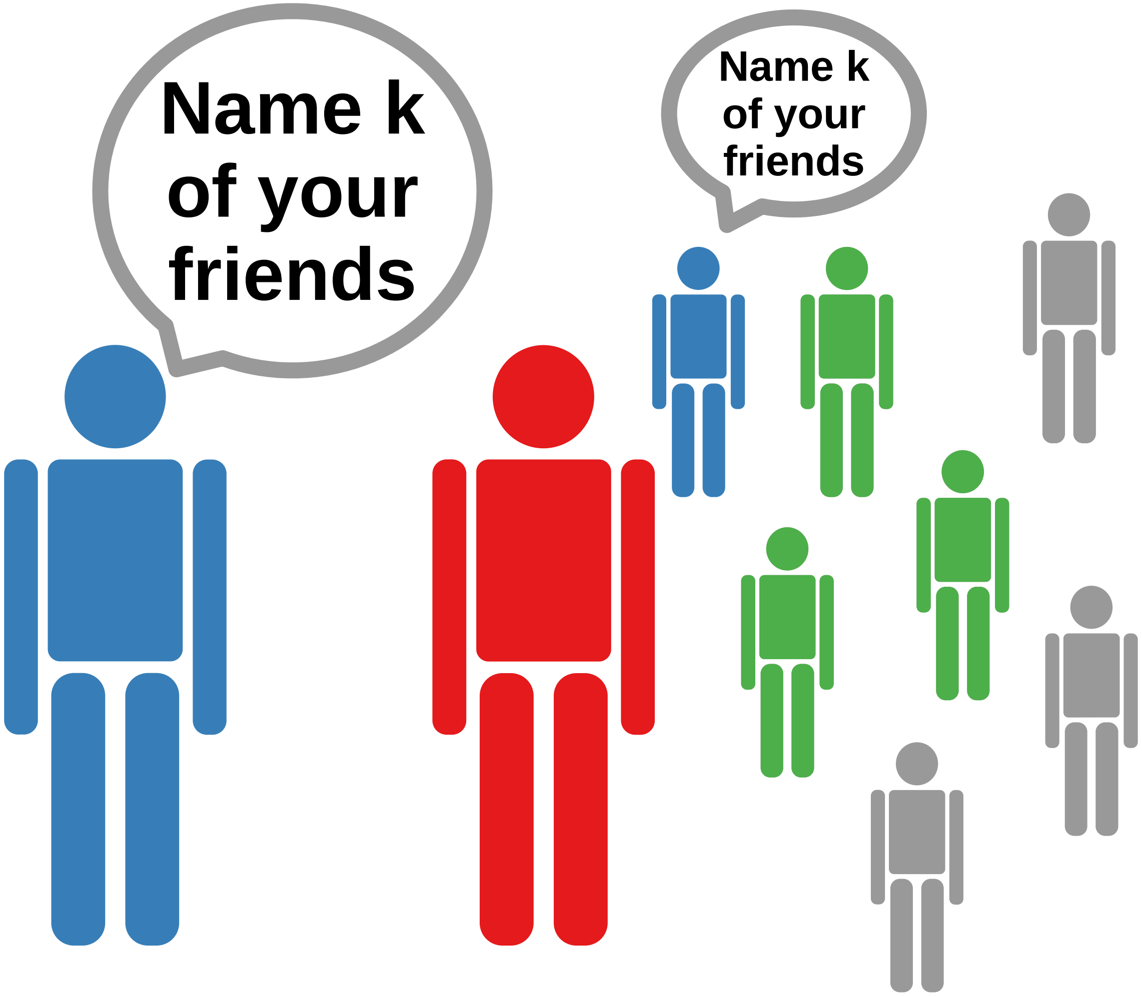
Snowball



Snowball: Advantages

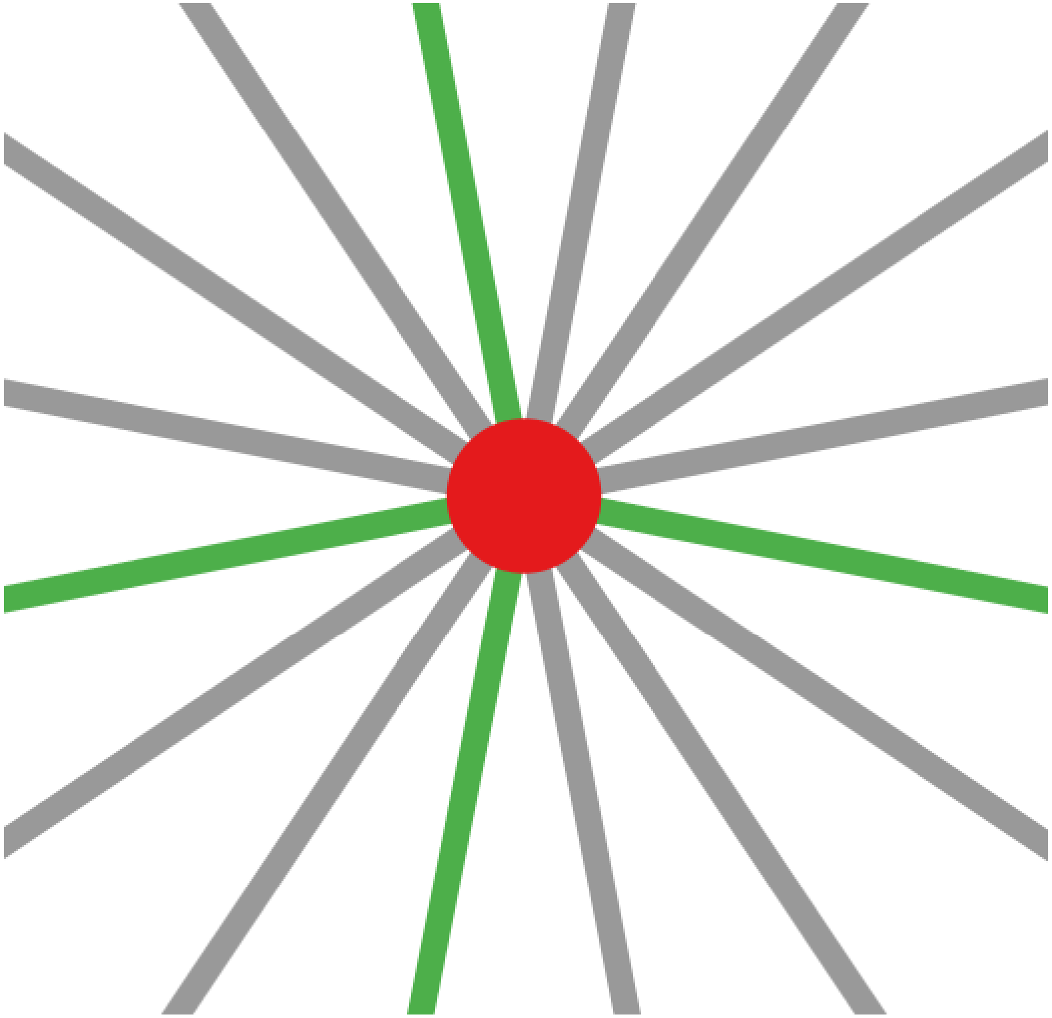
Snowball: Advantages

- Cheap in the physical world



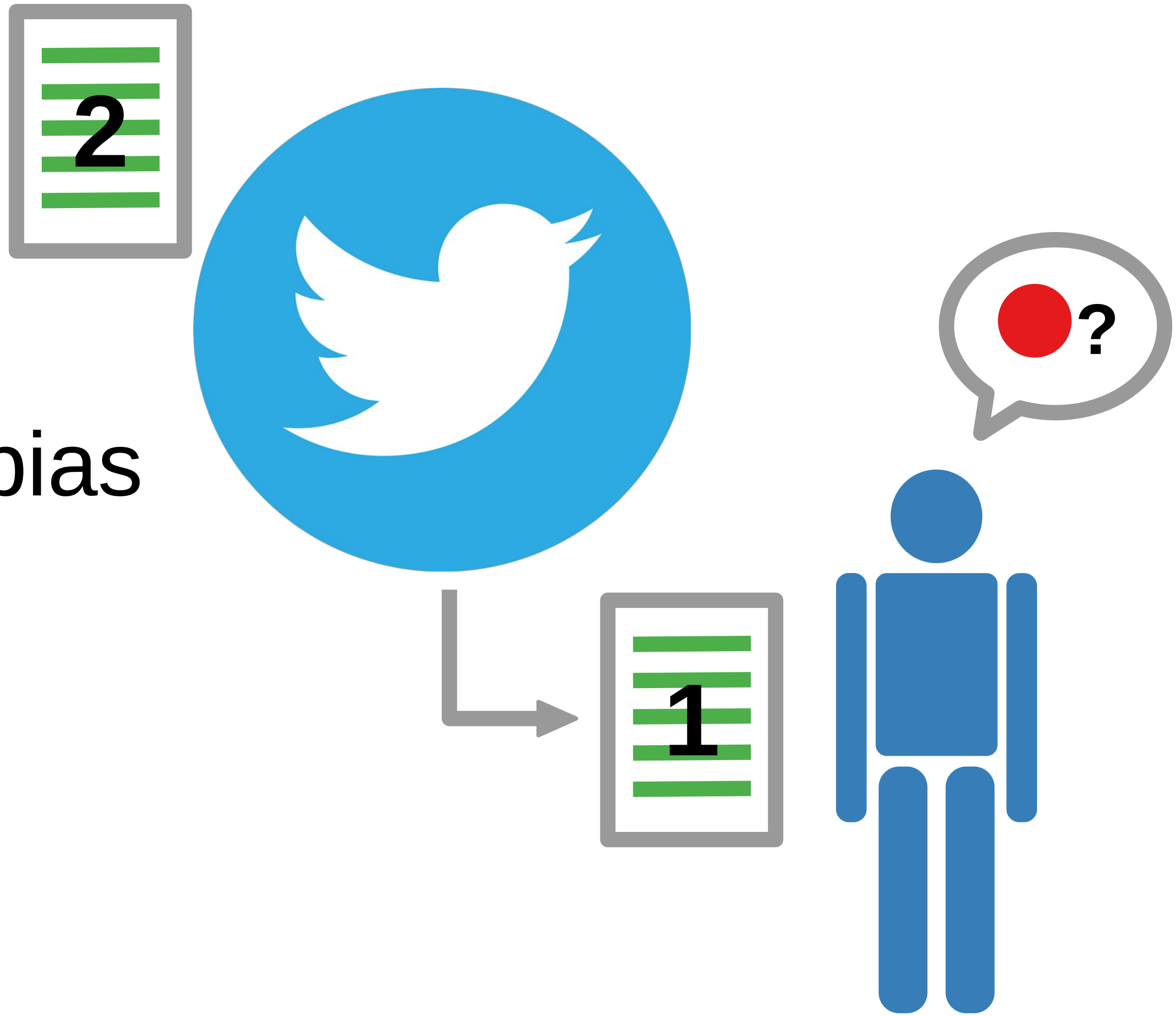
Snowball: Advantages

- Cheap in the physical world
- Smaller degree bias

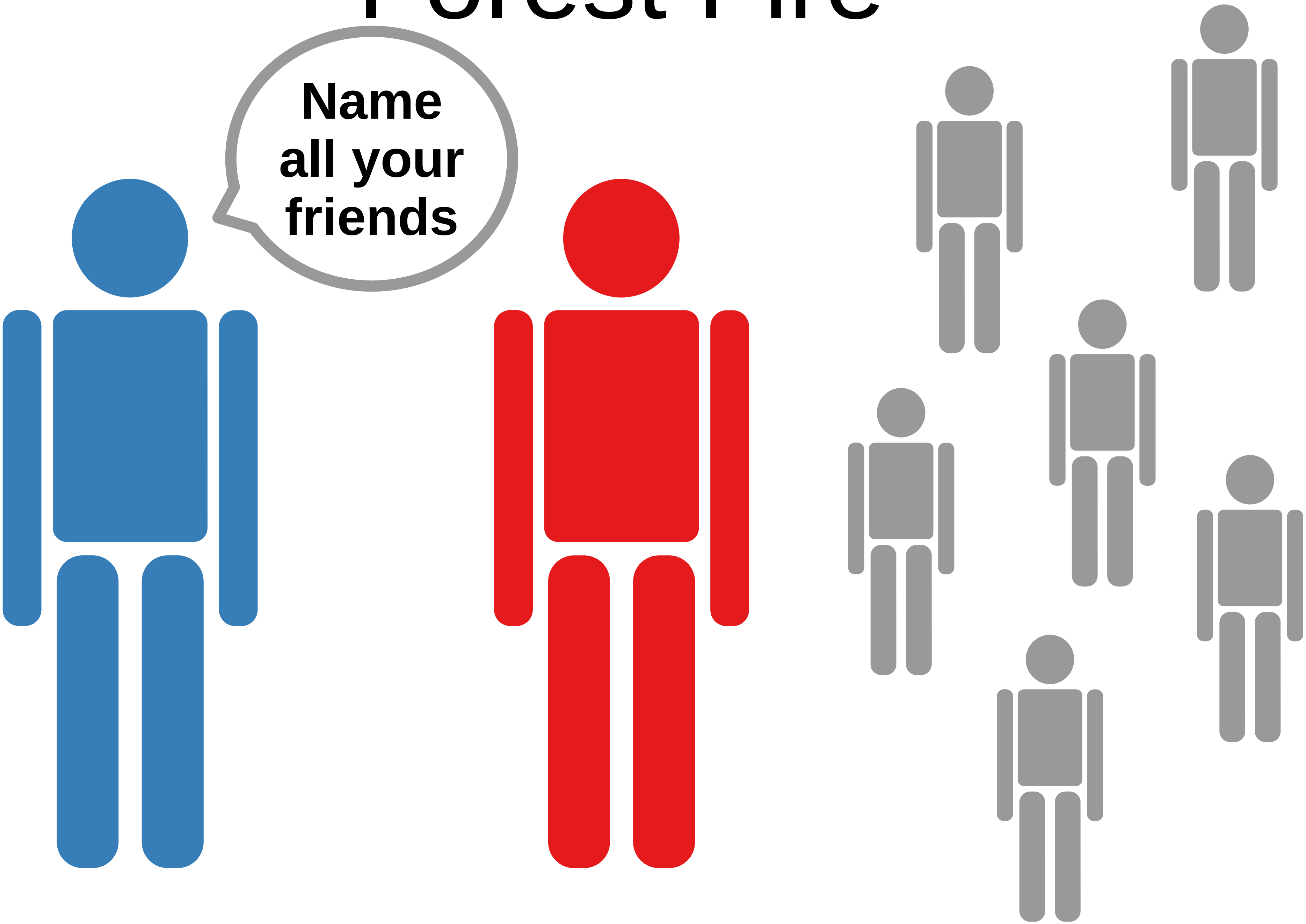


Snowball: Advantages

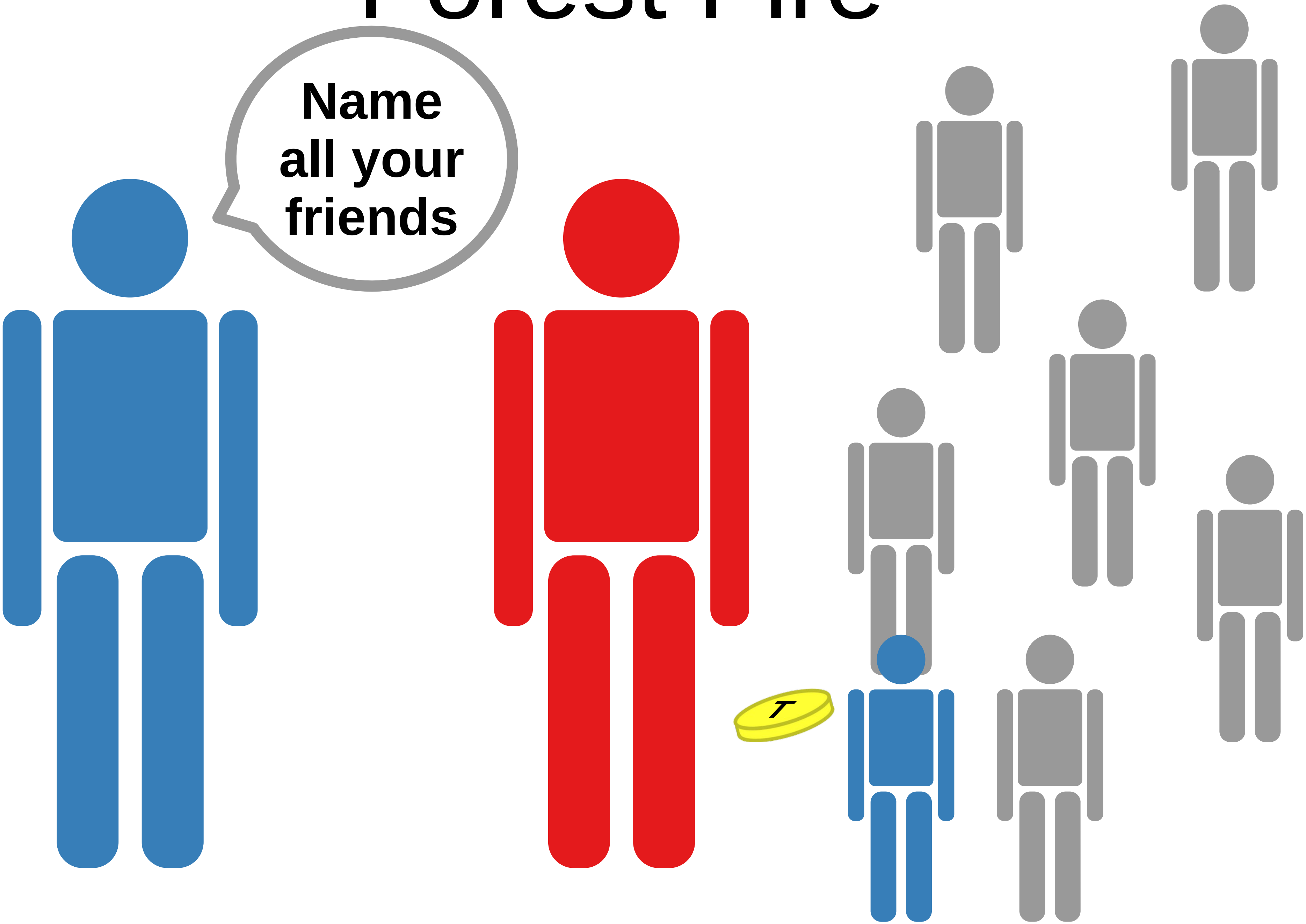
- Cheap in the physical world
- Smaller degree bias
- Works well with pagination



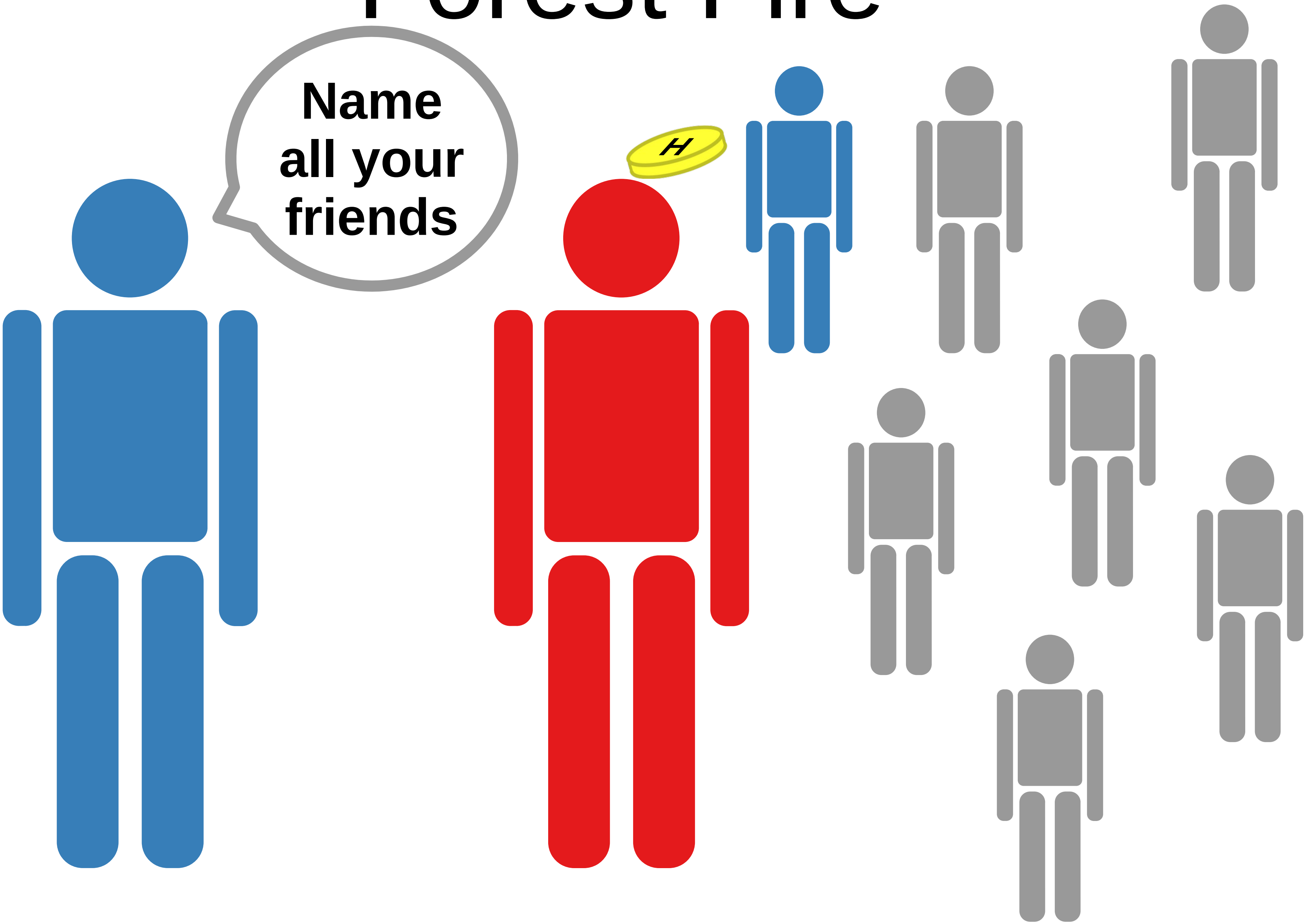
Forest Fire



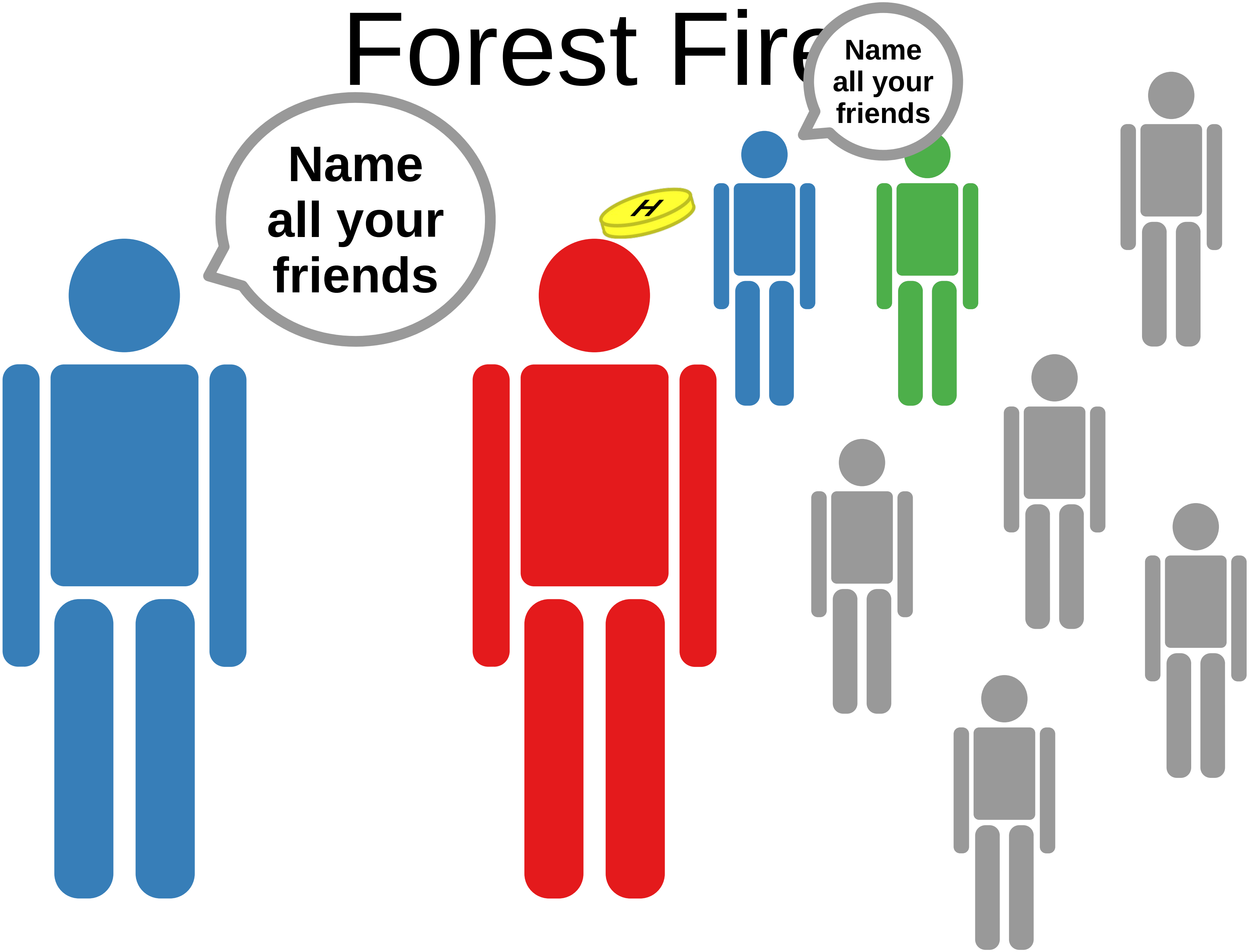
Forest Fire



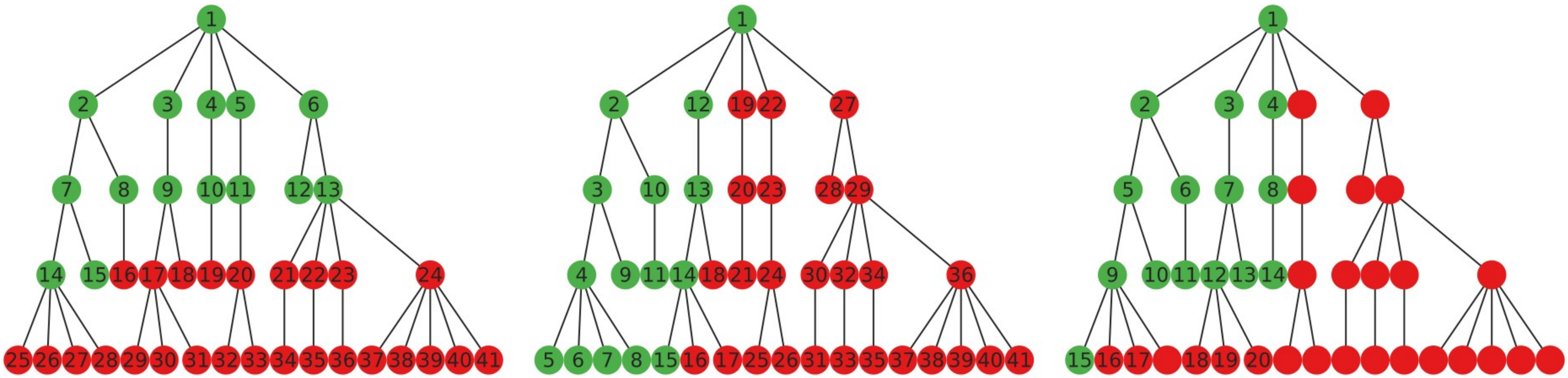
Forest Fire



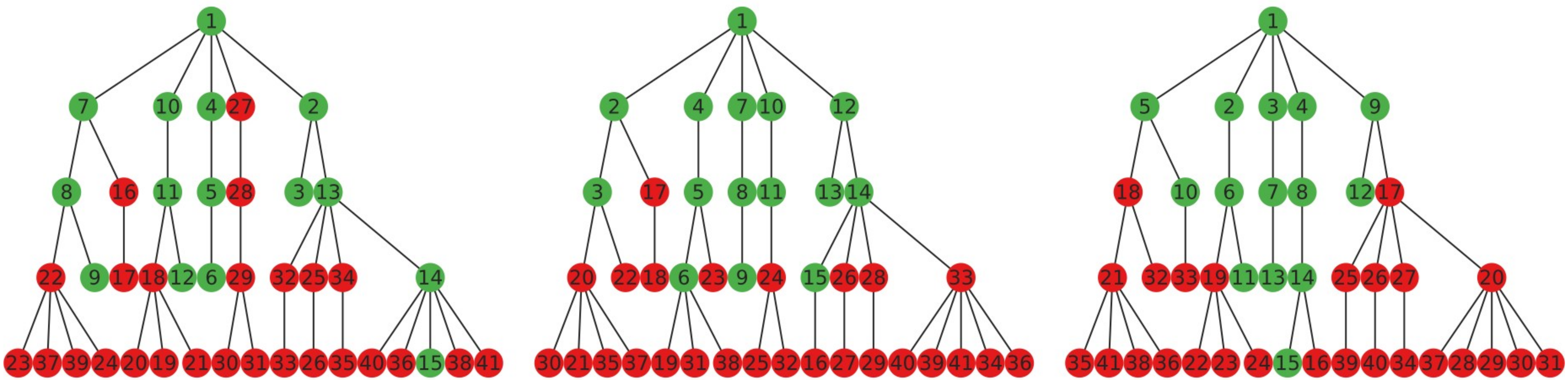
Forest Fire



The Network Sampling Zoo



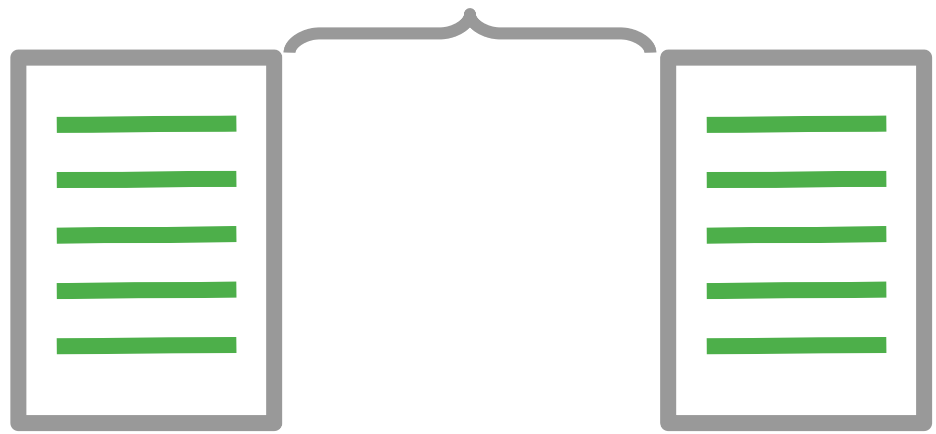
(a) BFS (b) DFS (c) Snowball



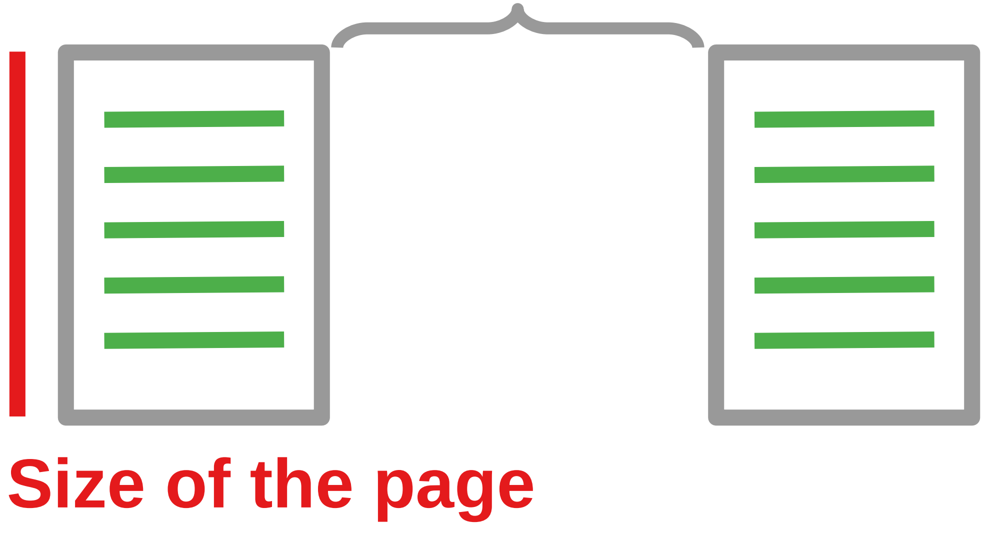
(d) Random Walk (e) MHRW (f) Forest Fire

(ii) Are there understudied real world obstacles that should make us reconsider how we choose the best sampling strategy?

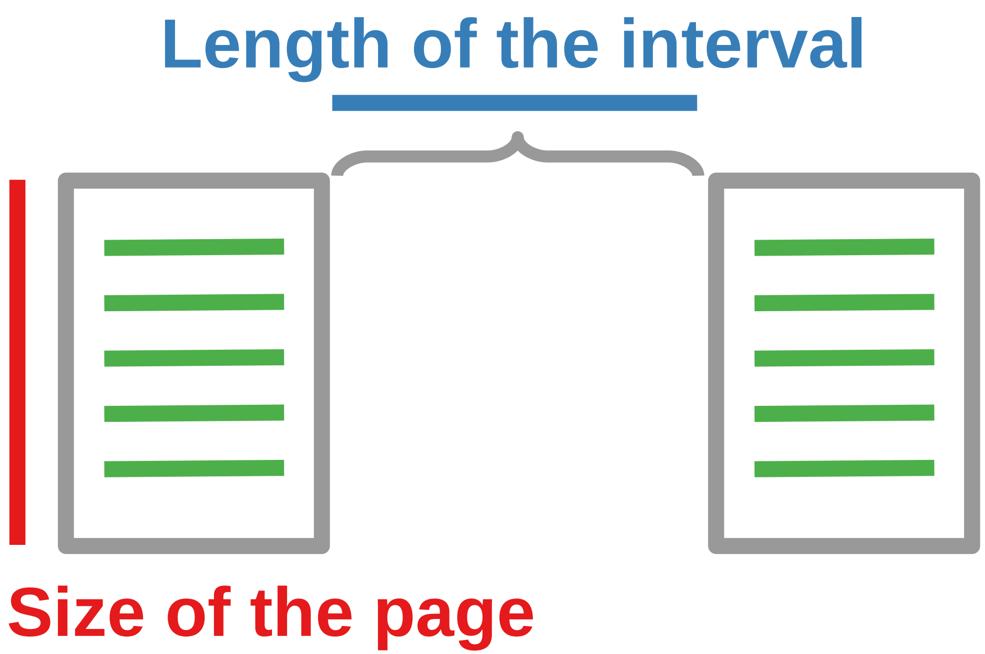
Social Media APIs



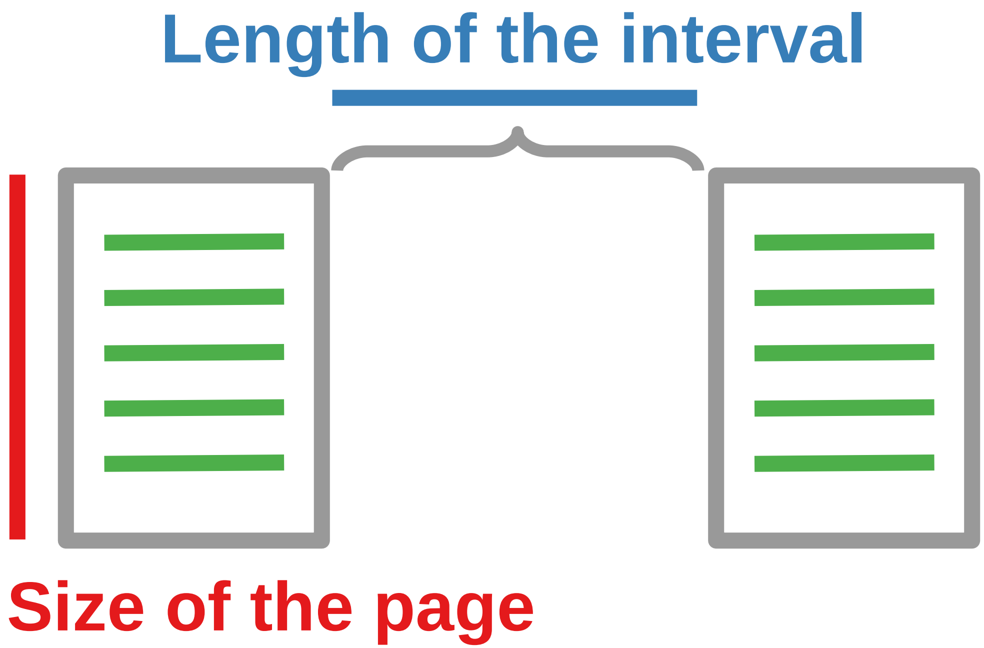
Social Media APIs



Social Media APIs



Social Media APIs



(...latency)

Pagination Paradox

Pagination Paradox

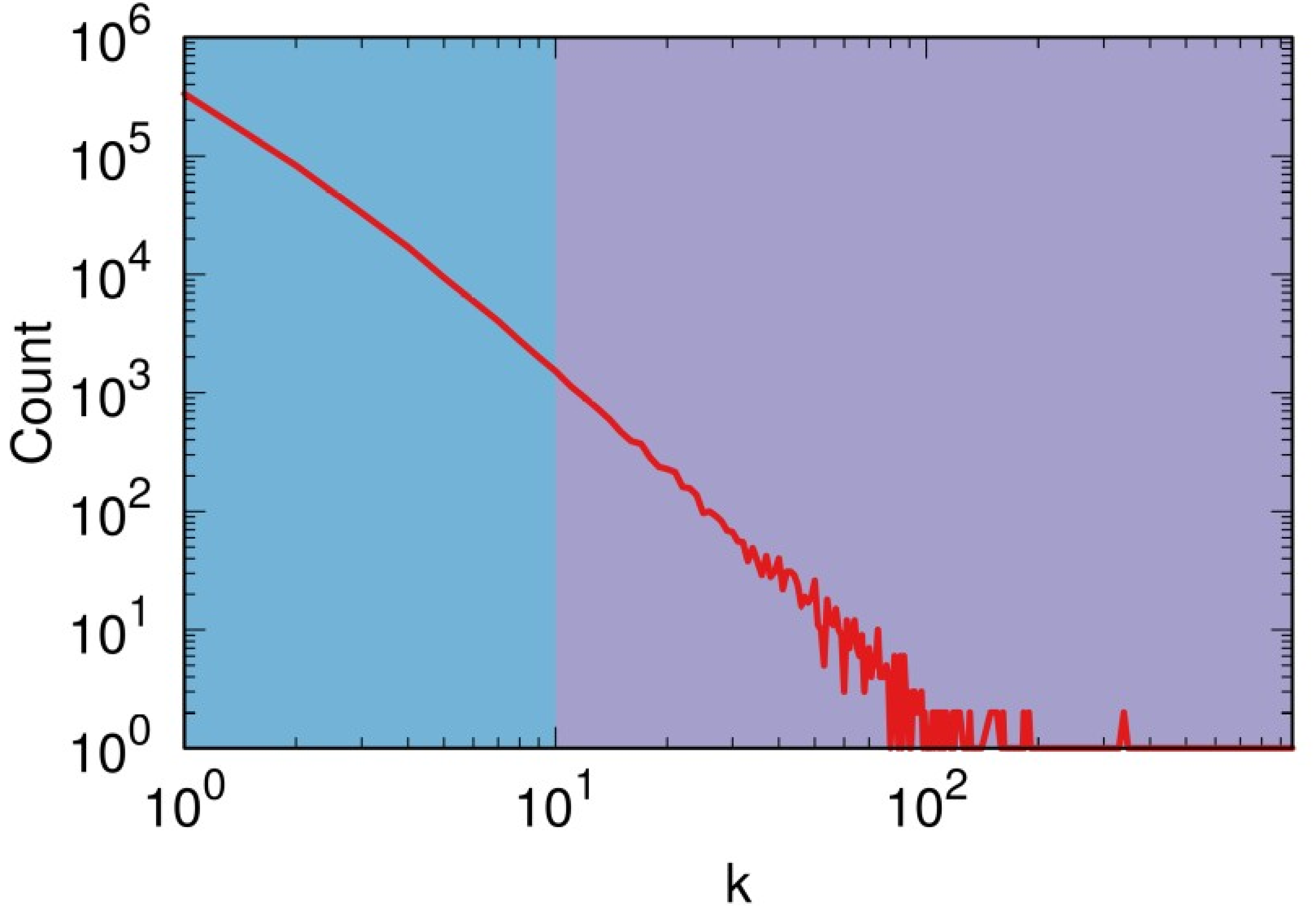
- Edges per page: 100
- Seconds between queries: 2
- 50 edges / sec

Pagination Paradox

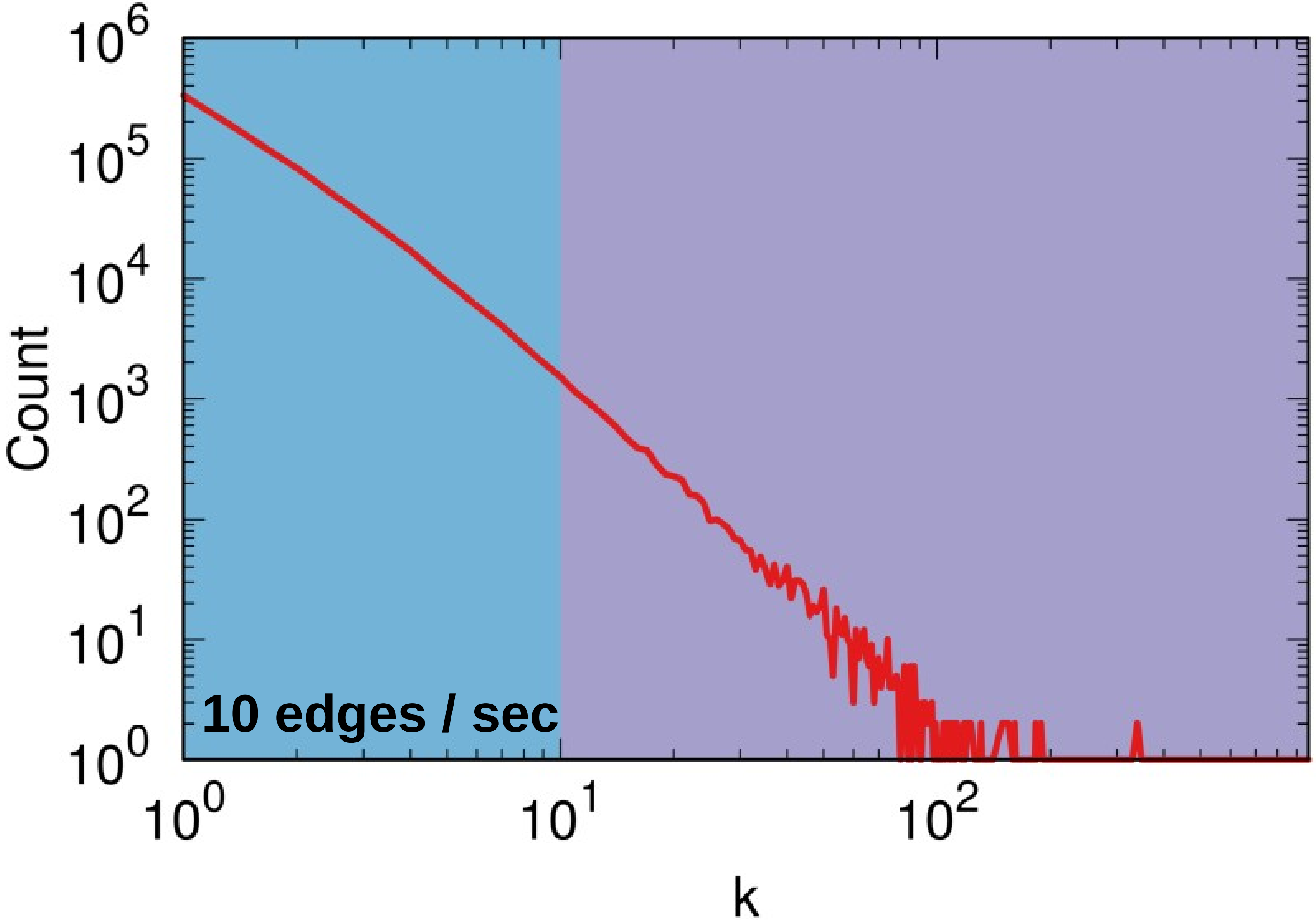
- Edges per page: 100
- Seconds between queries: 2
- 50 edges / sec

- Edges per page: 10
- Seconds between queries: 1
- 10 edges / sec

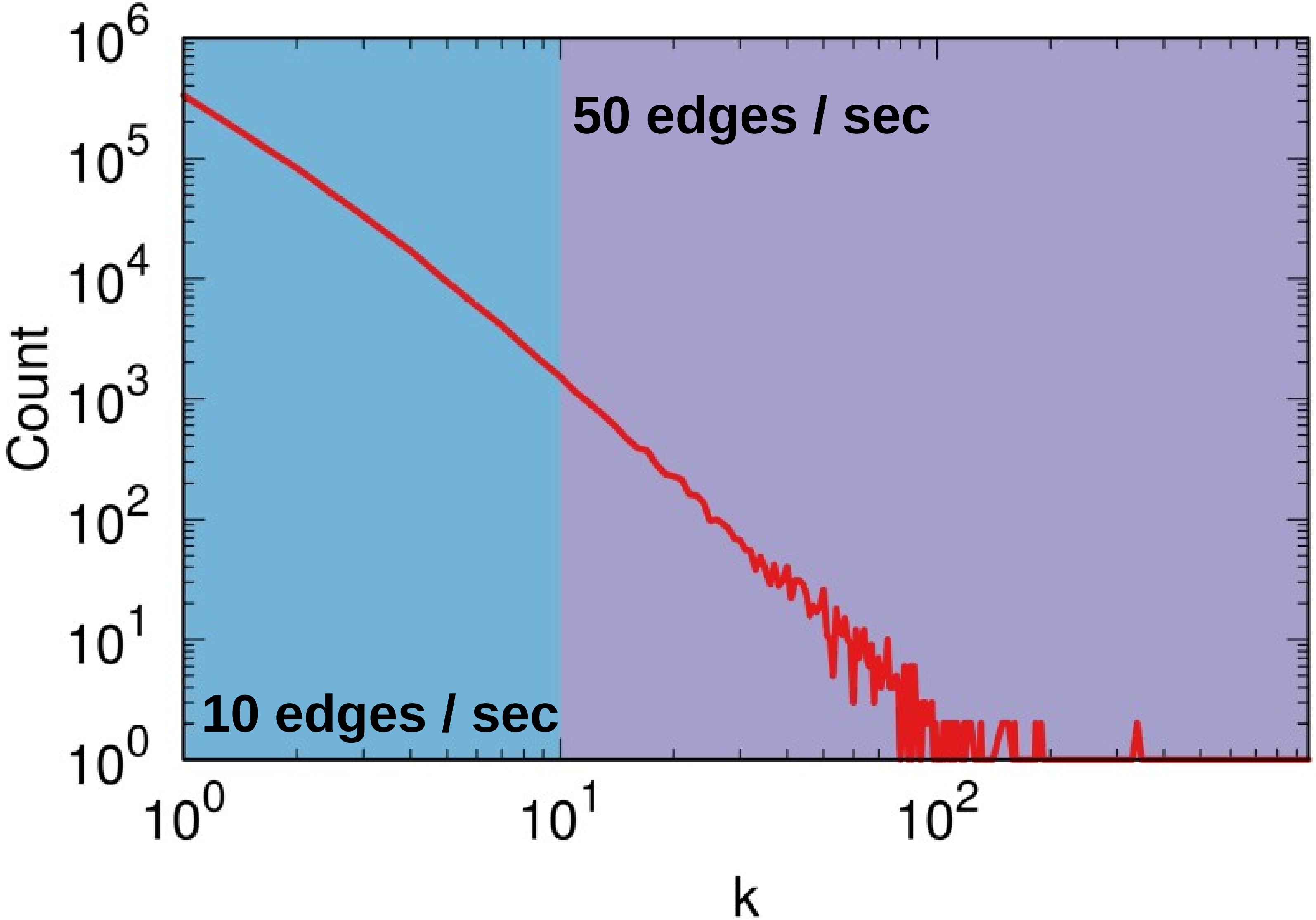
Pagination Paradox



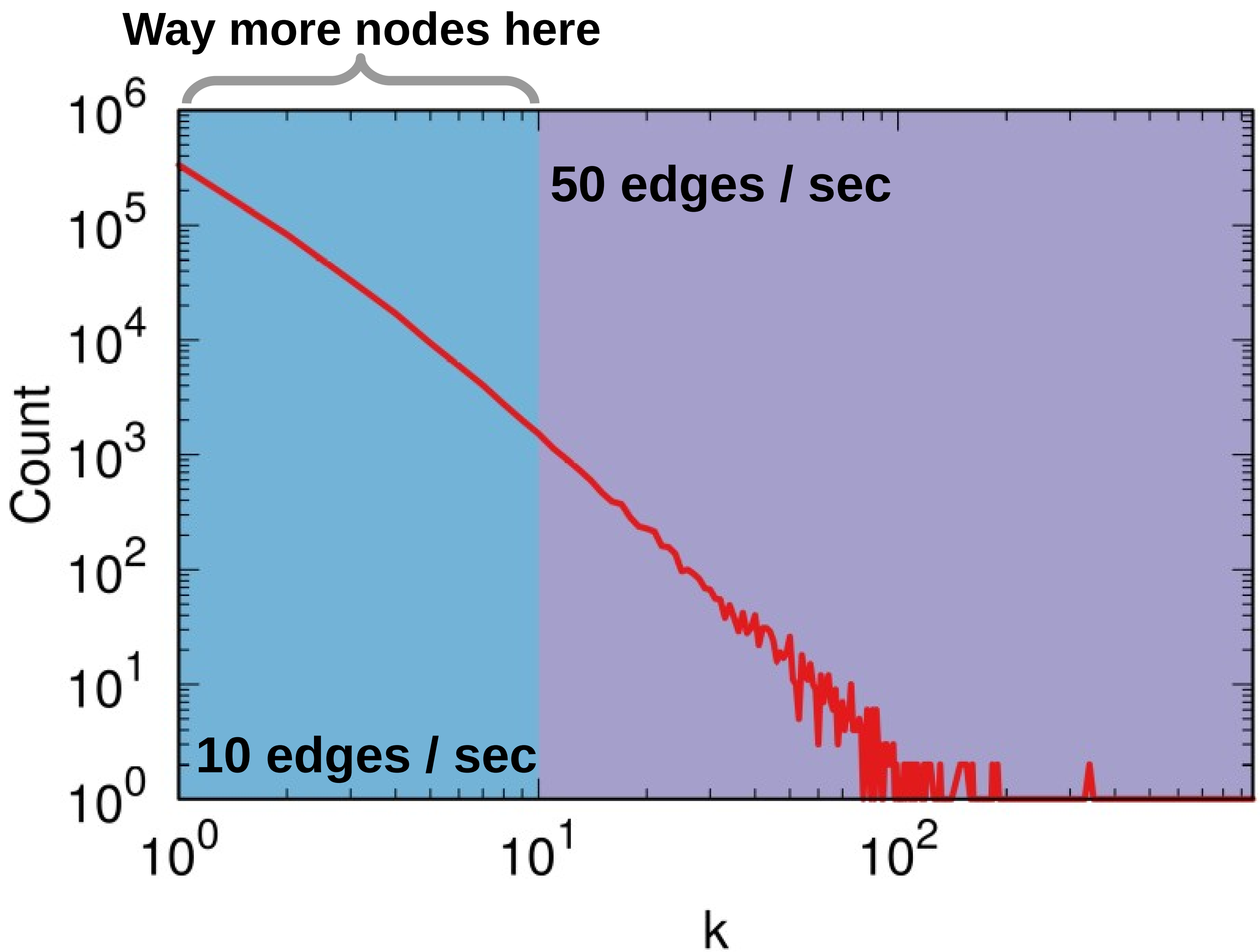
Pagination Paradox



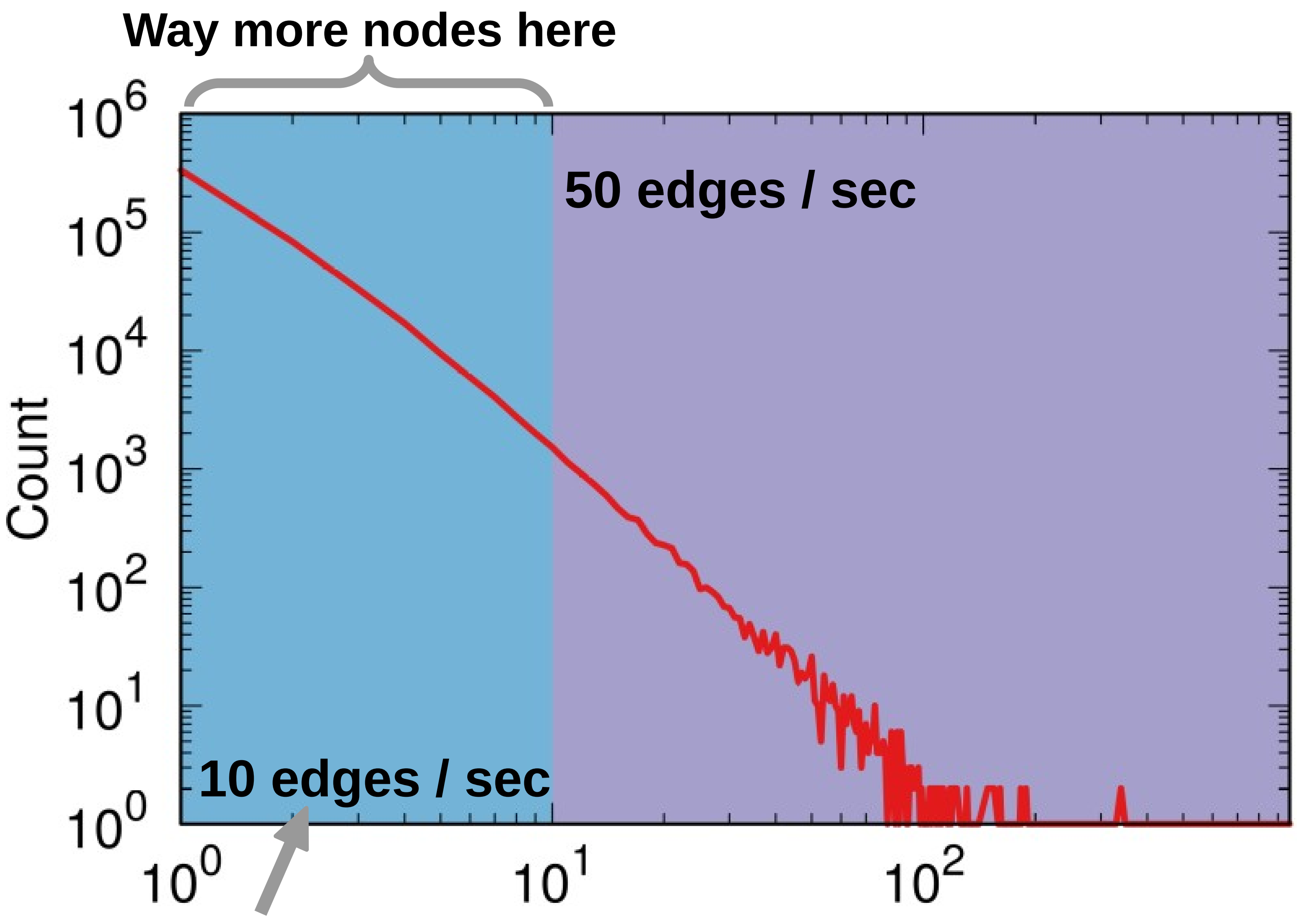
Pagination Paradox



Pagination Paradox



Pagination Paradox



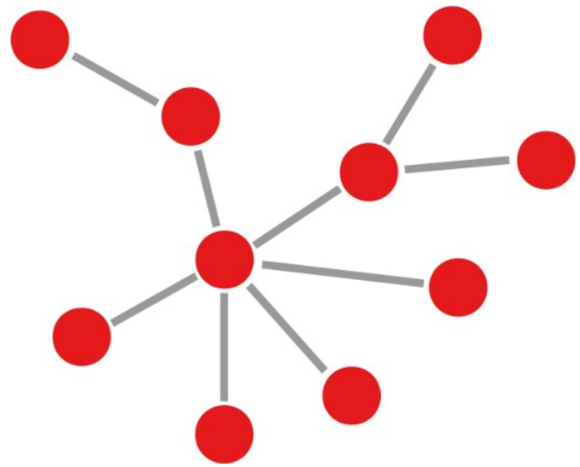
5x slower in theory
2x faster in practice

Benchmark Setup

- Three types of topologies:
  - Barabasi-Albert
  - Small World
  - LFR Benchmark

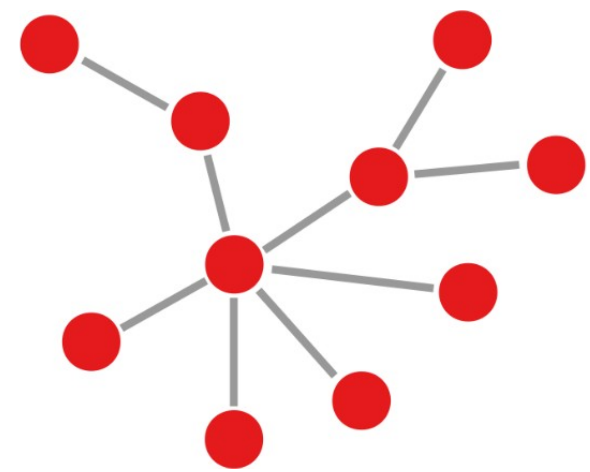
Benchmark Setup

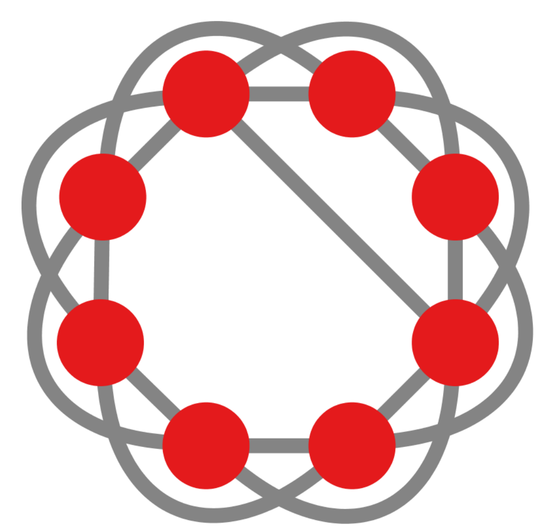
- Three types of topologies:
  - Barabasi-Albert
  - Small World
  - LFR Benchmark



Benchmark Setup

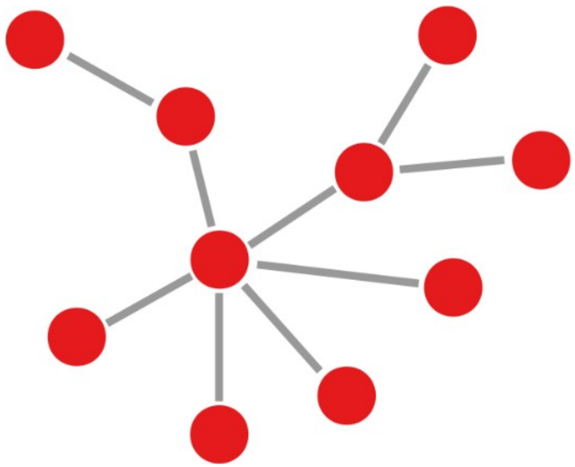
- Three types of topologies:
  - Barabasi-Albert
  - Small World
  - LFR Benchmark

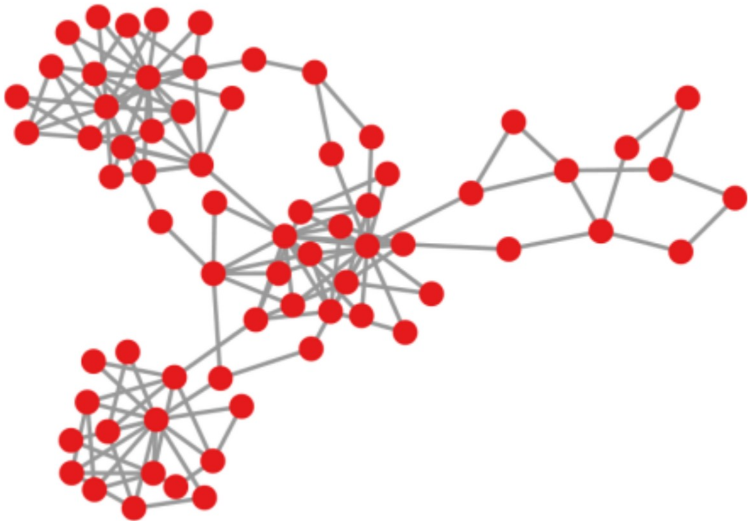


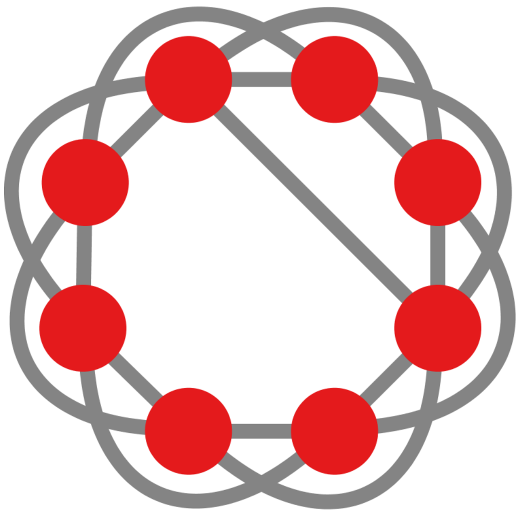


Benchmark Setup

- Three types of topologies:
  - Barabasi-Albert
  - Small World
  - LFR Benchmark







Benchmark Setup

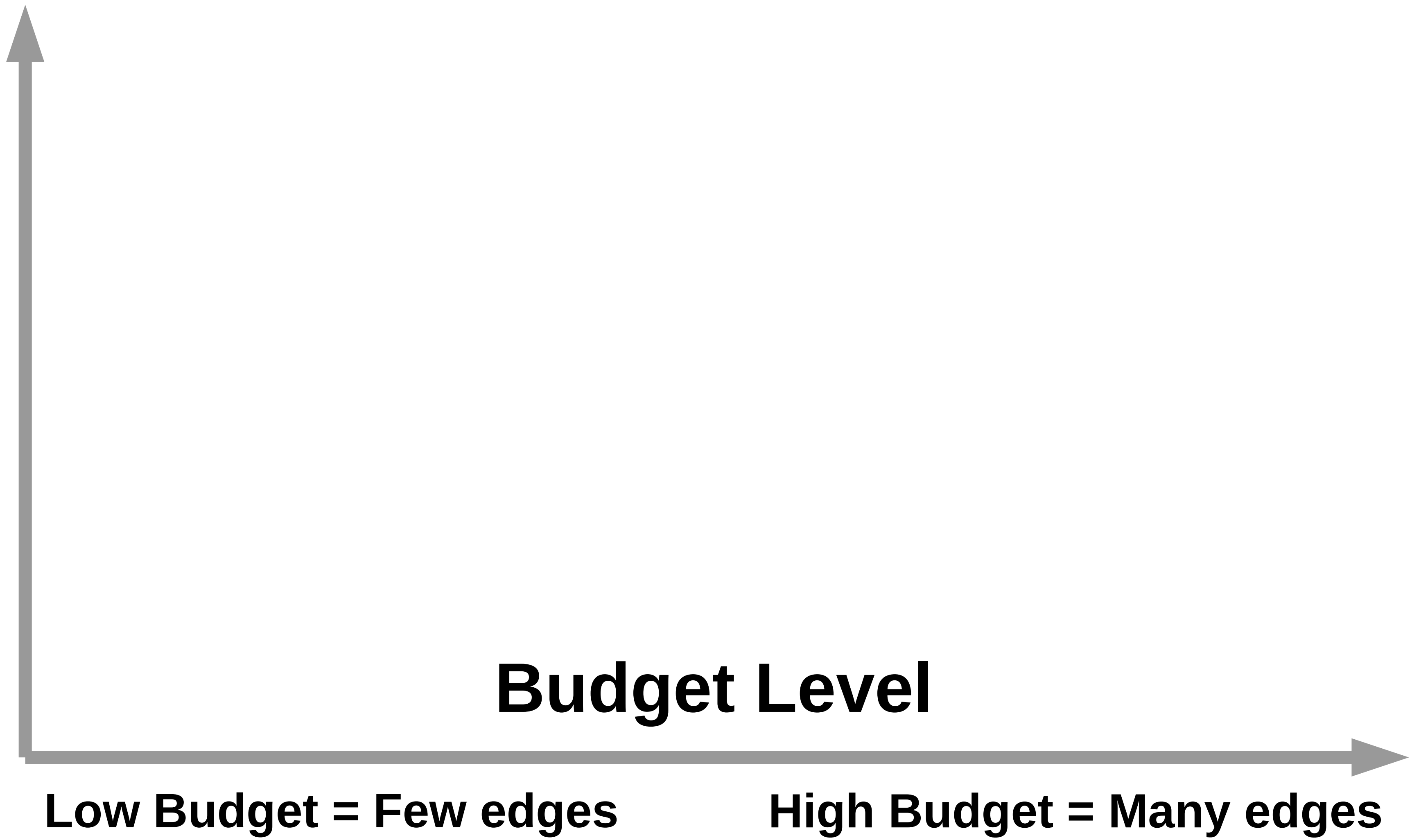
- Six API systems from real social media:
  - Flickr
  - Lastfm
  - Twitter
  - Youtube
  - Tumblr
  - Google+

Benchmark Setup

- Different objectives:
  - Degree Distribution
  - Assortativity / Disassortativity
  - Centrality
  - Reciprocity

Benchmark Setup

Benchmark Setup



Benchmark Setup

Quality Measure

(NB: not always “lower is better”)

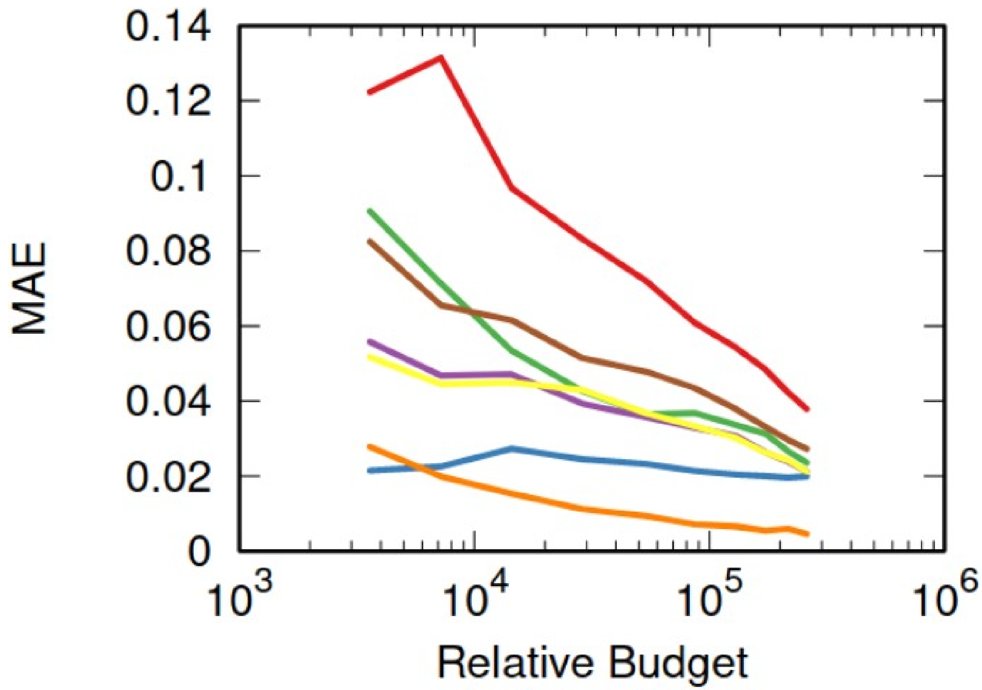
Budget Level

Low Budget = Few edges **High Budget = Many edges**

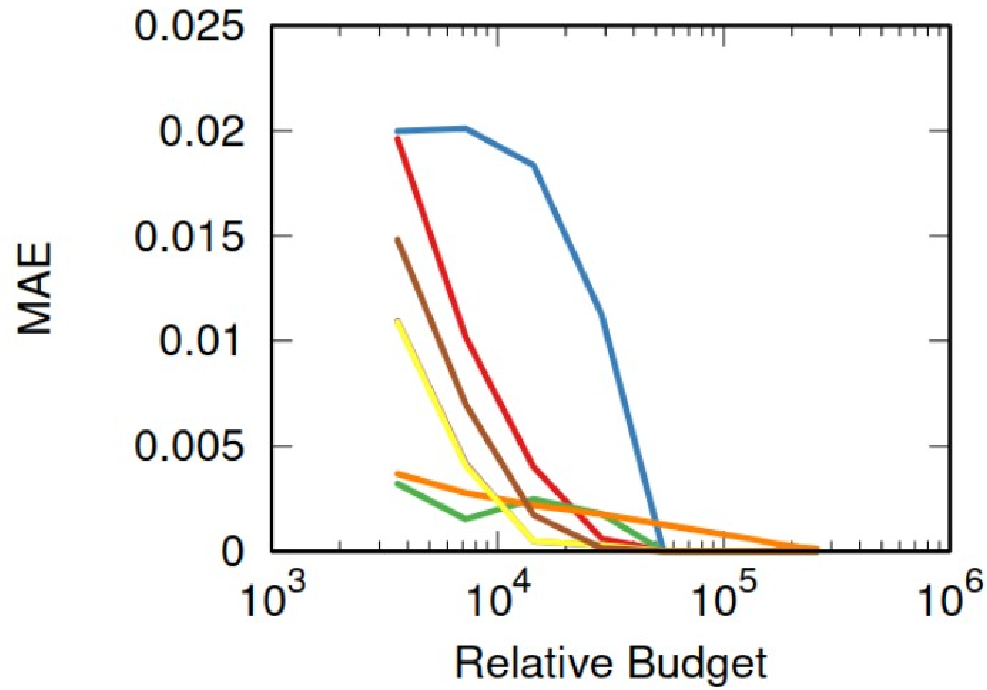
Disassortativity MAE

(lower is better)

Tumblr



LastFM

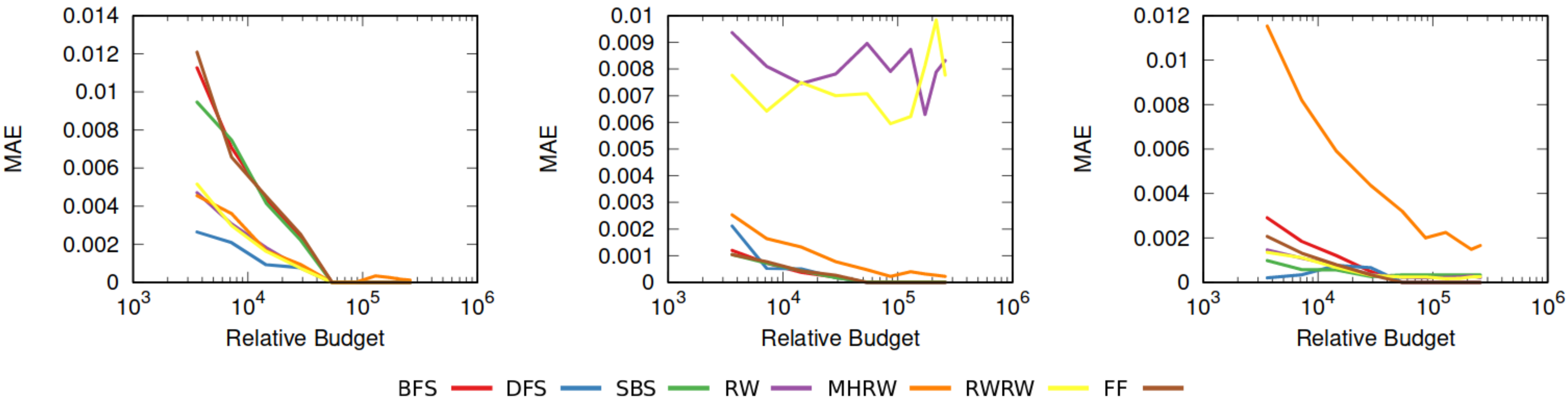


BFS DFS SBS RW MHRW RWRW FF

Assortativity MAE

(lower is better)

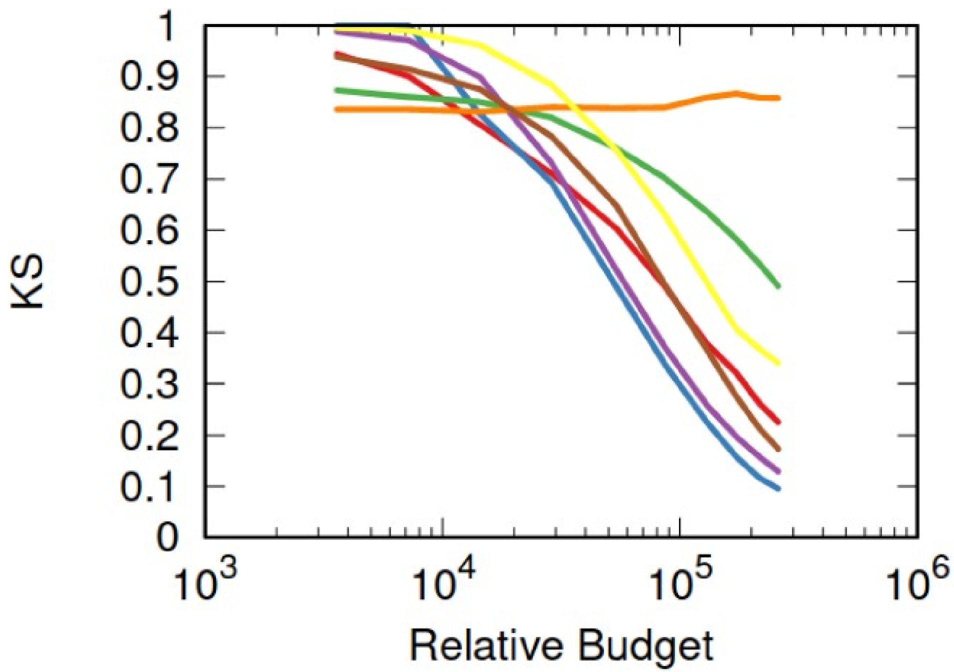
SW **B-A** **LFR**



Budget Levels

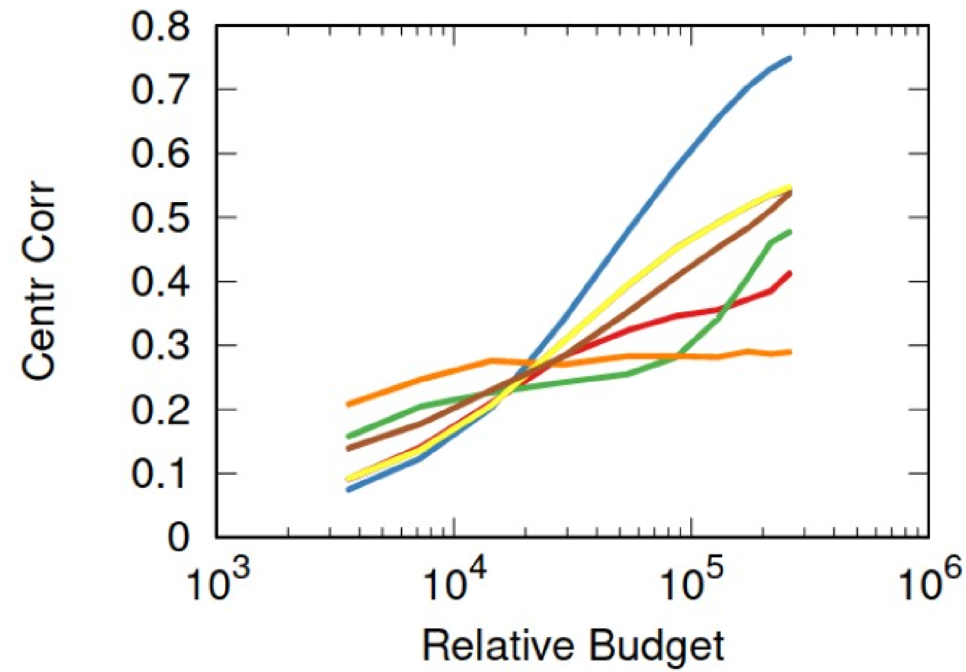
Degree Distribution

(lower is better)



Centrality Correlation

(higher is better)



BFS DFS SBS RW MHRW RWRW FF

Conclusion

Conclusion

- We have to sample

Conclusion

- We have to sample
- We have good theory…

Conclusion

- We have to sample
- We have good theory…
- ...for the case of infinite time and paging sizes

Conclusion

- We have to sample
- We have good theory…
- ...for the case of infinite time and paging sizes
- Which is not realistic

Conclusion

- We have to sample
- We have good theory…
- ...for the case of infinite time and paging sizes
- Which is not realistic
- Realistic constraints paint a critical picture

Thanks

Benchmarking API Costs of Network Sampling Strategies

Michele Coscia & Luca Rossi
mcos@itu.dk lucr@itu.dk
http://www.michelecoscia.com