Discovering Hidden Groups in Communication Networks

Jeffrey Baumes
Mark Goldberg
Malik Magdon-Ismail
William Wallace

What is a Hidden Group?

- Actors in a social network form groups.
- Some groups try to hide their communications in the background.
- How do we discover such *hidden* groups?

How to Find Hidden Groups

- Individual (semantic) analysis
- Automated structural/statistical analysis

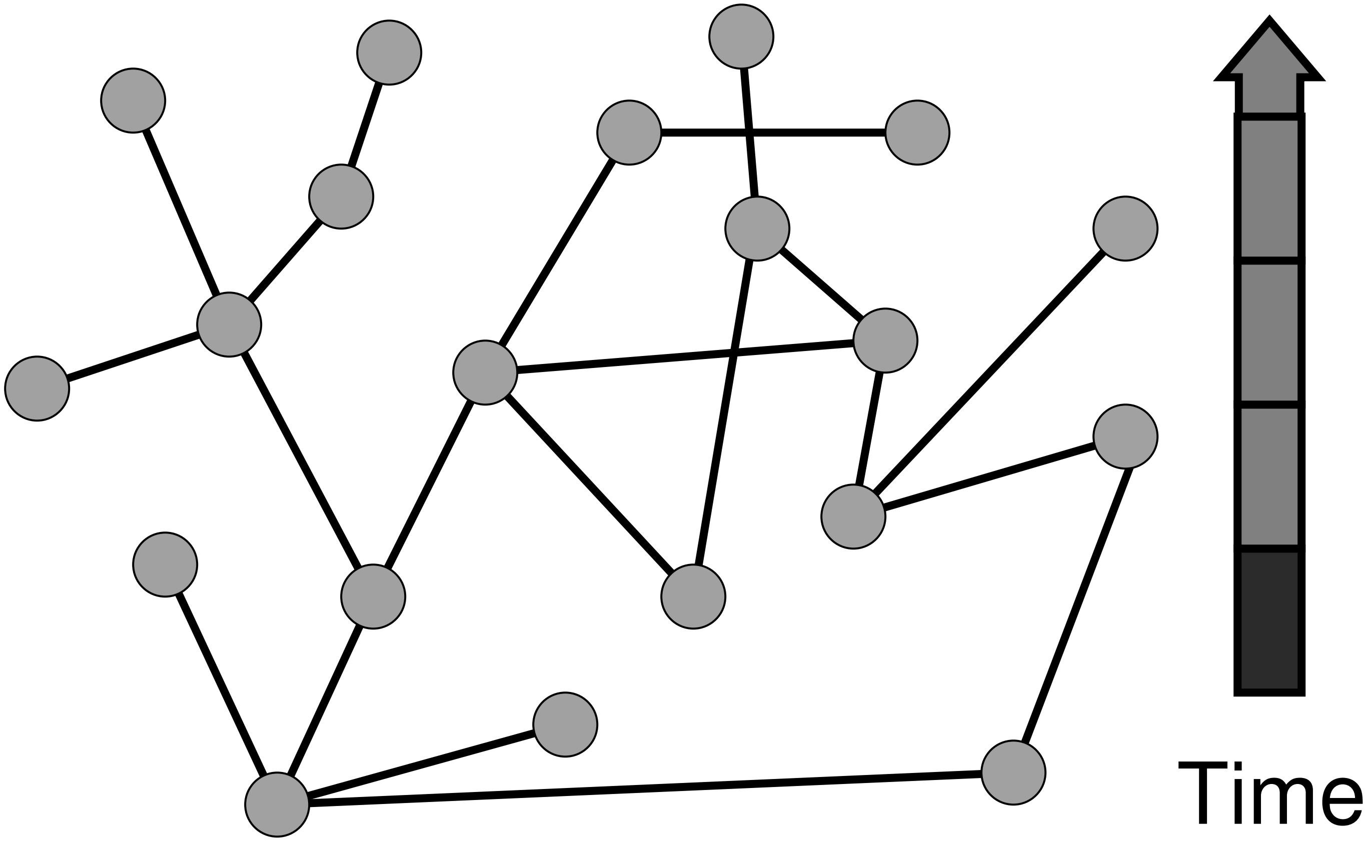
100 actor society

10^{30} groups

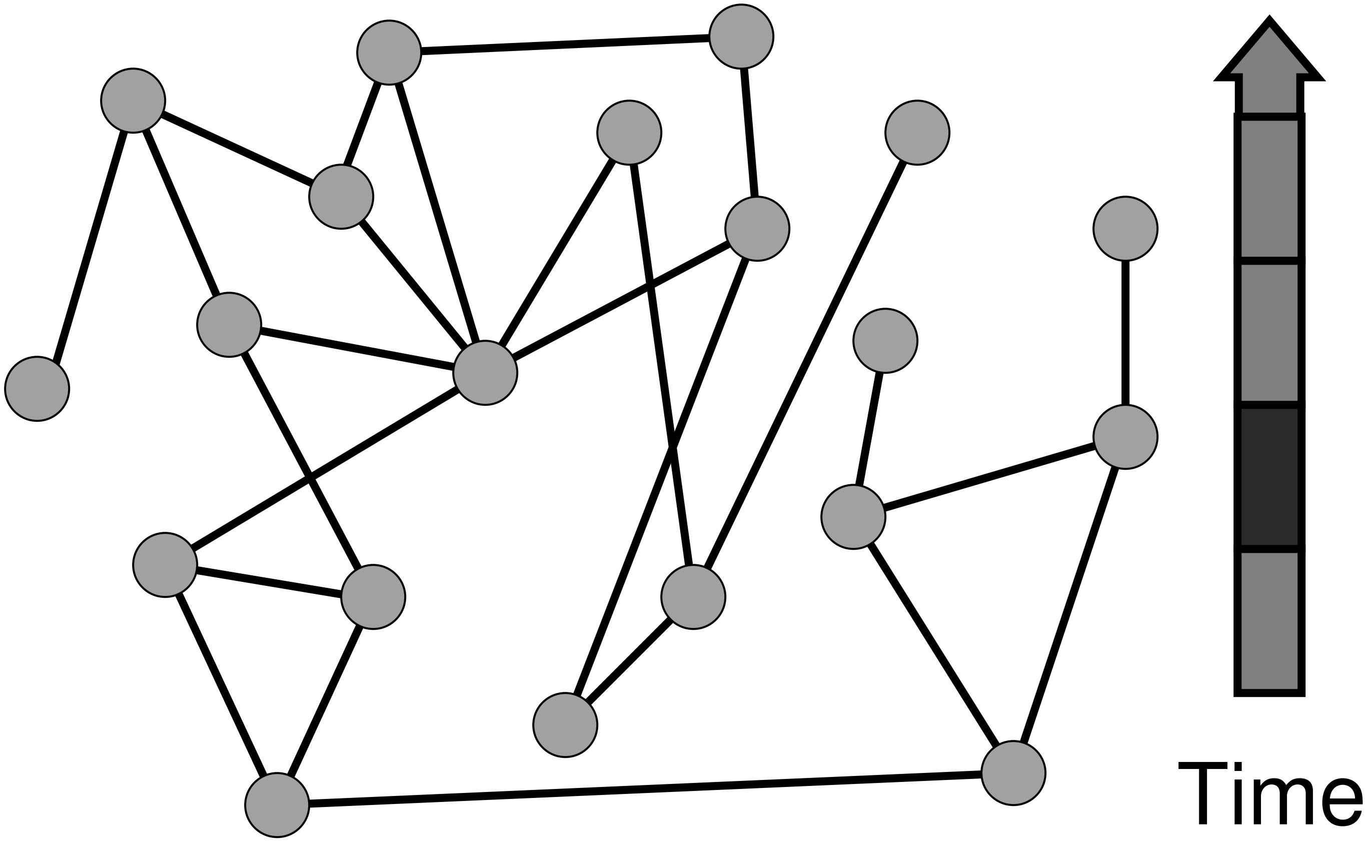
How to Find Hidden Groups

- Need to preprocess the network based on *structure alone*
- *Efficiently!*

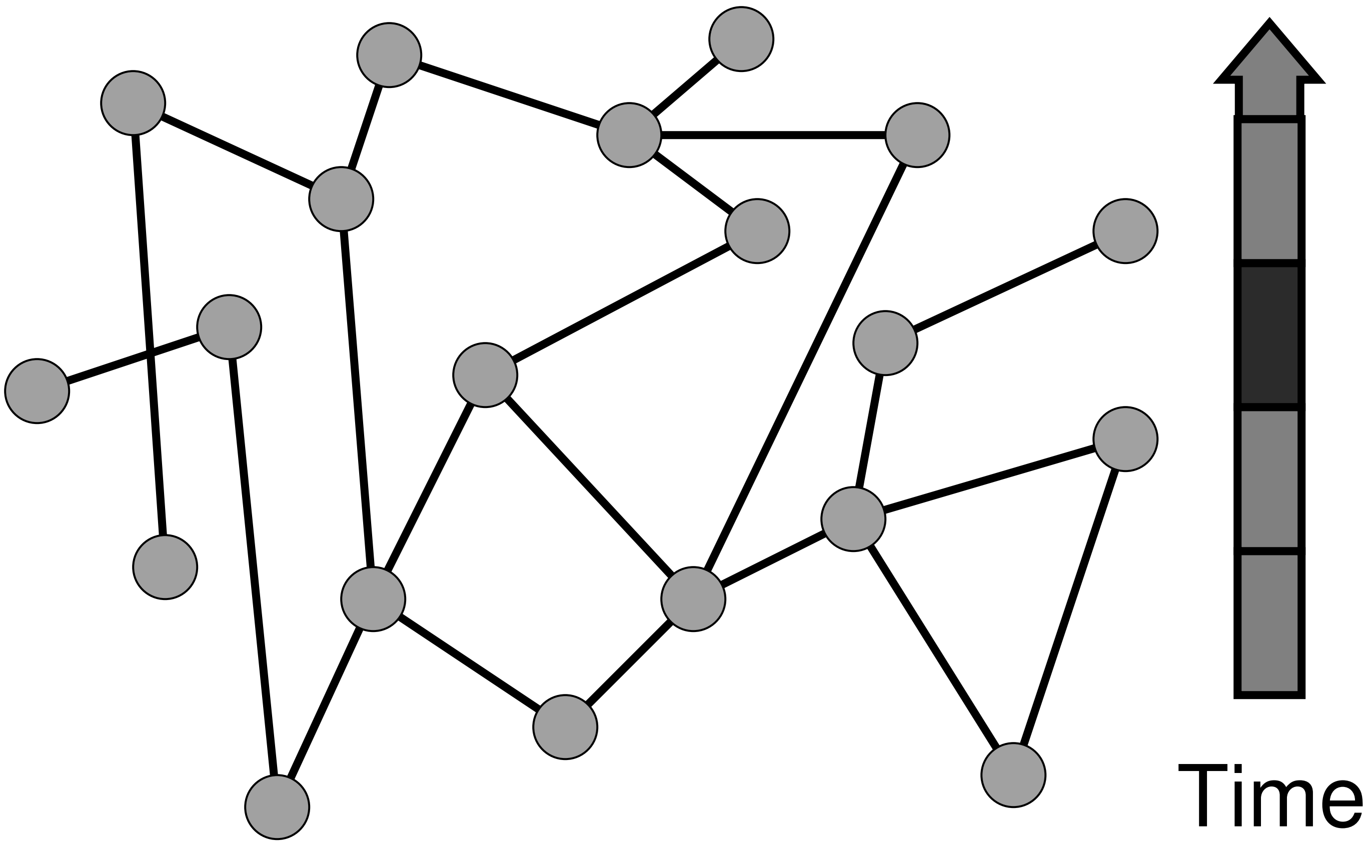
Which is the Hidden Group



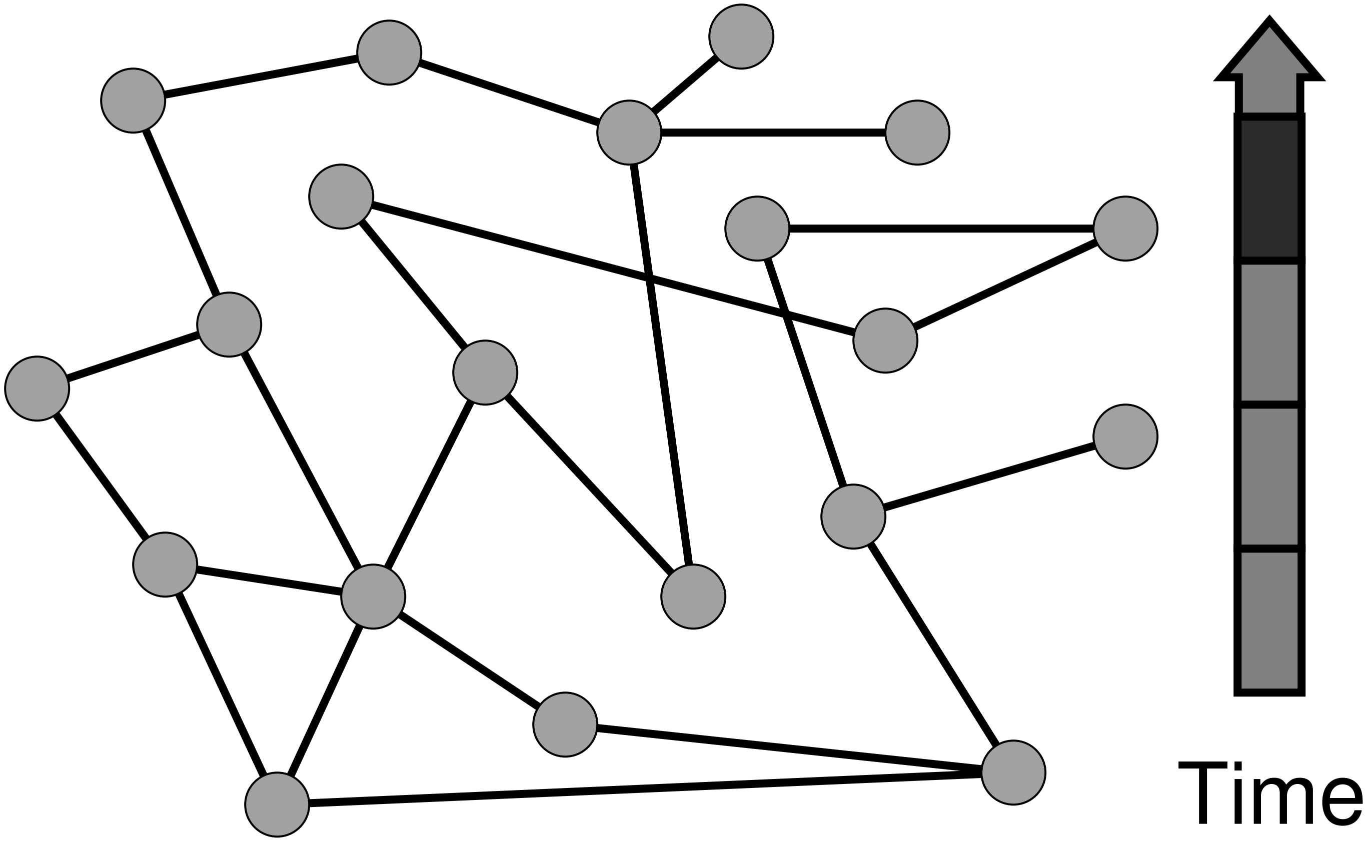
Which is the Hidden Group



Which is the Hidden Group



Which is the Hidden Group



Rensselaer

Goal

- Find a communication pattern to extract hidden group from background
- Design efficient algorithm
- Develop efficient implementation

Overview

- Hidden group communication patterns
- Efficient discovery algorithm
- Background communication models
- Simulation results
- Conclusions

Overview

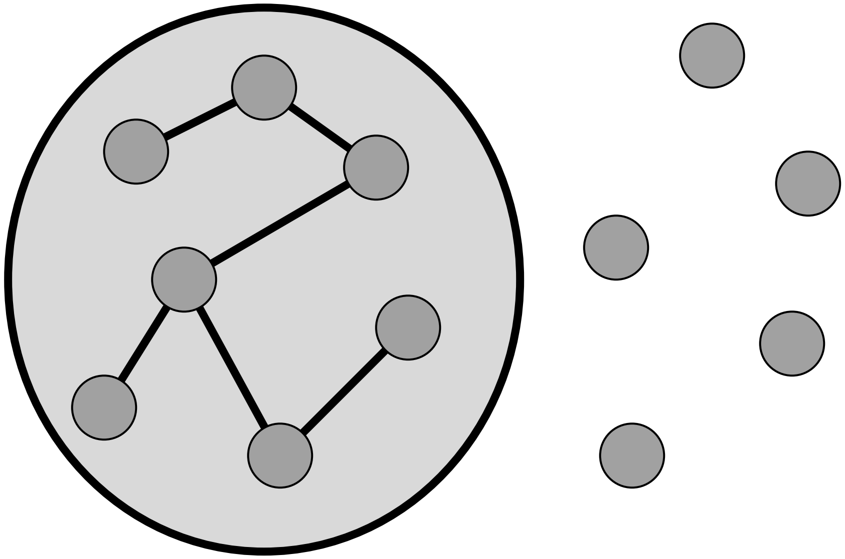
- Hidden group communication patterns
- Efficient discovery algorithm
- Background communication models
- Simulation results
- Conclusions

Hidden Group Communication Pattern

- Assumption: group coordination within some time interval, *connected*
- Collect communications at this interval
- Distinguishing characteristic:
  - Hidden group connected in each of these networks, *persistently connected*

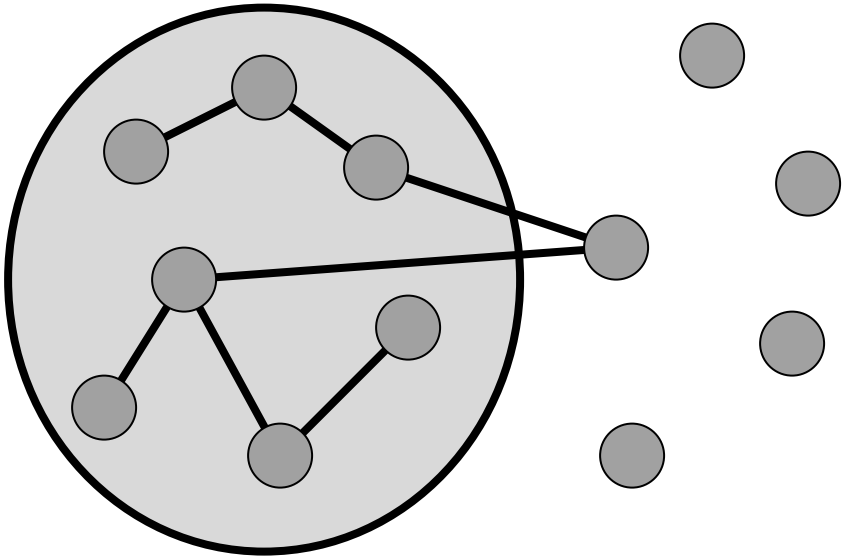
Internally Connected Groups

Internally connected (non-trusting) groups pass information internally

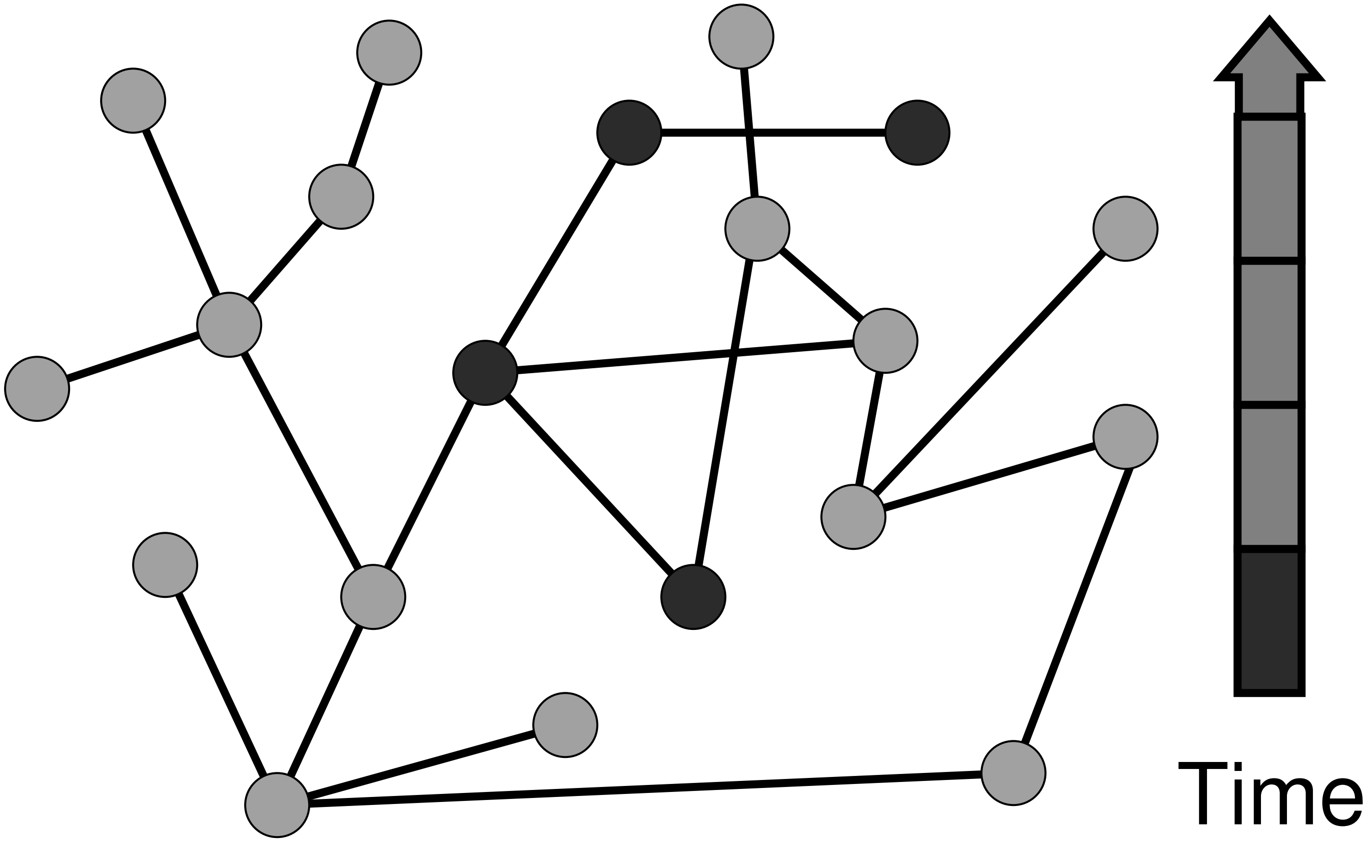


Externally Connected Groups

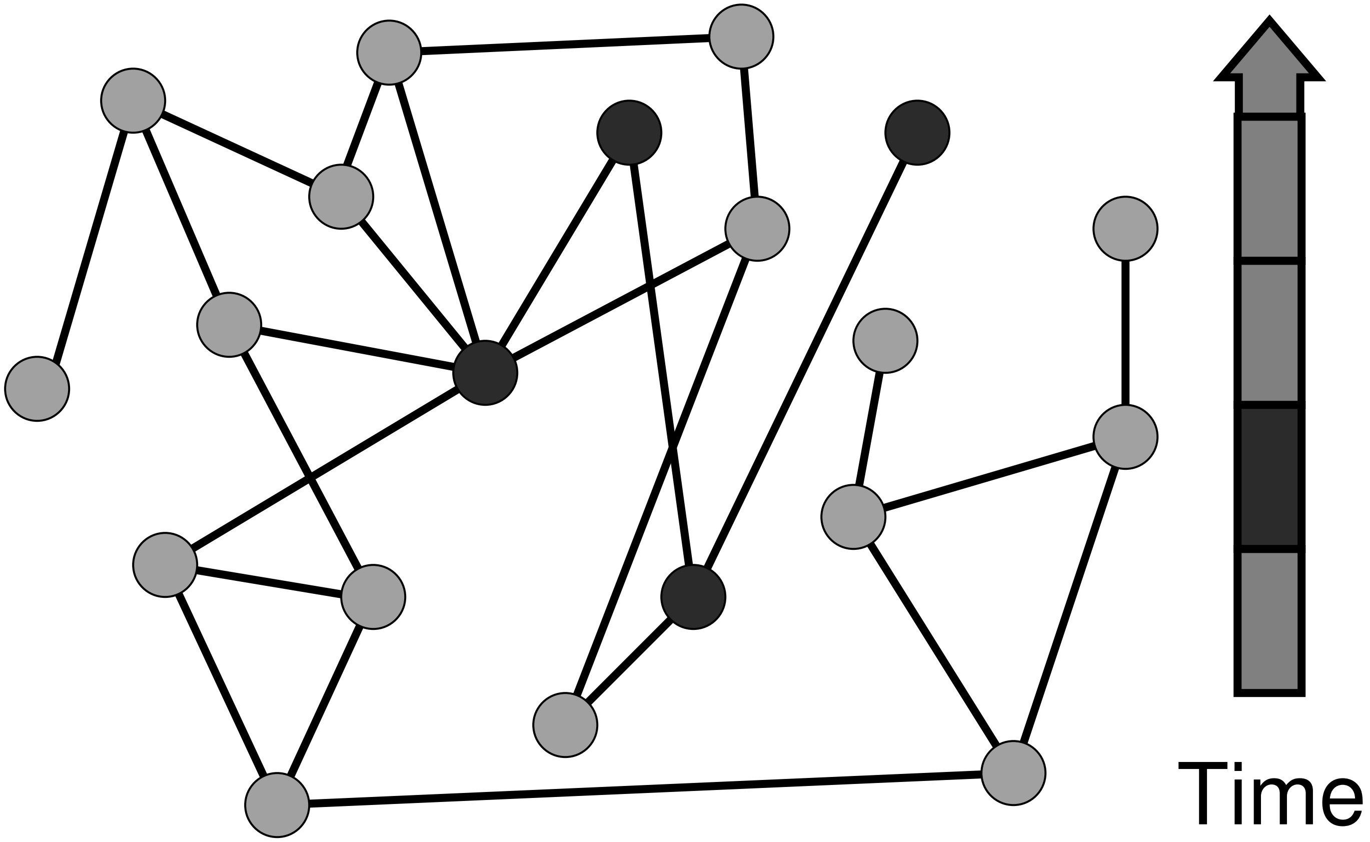
Externally connected (trusting) groups may use outside actors



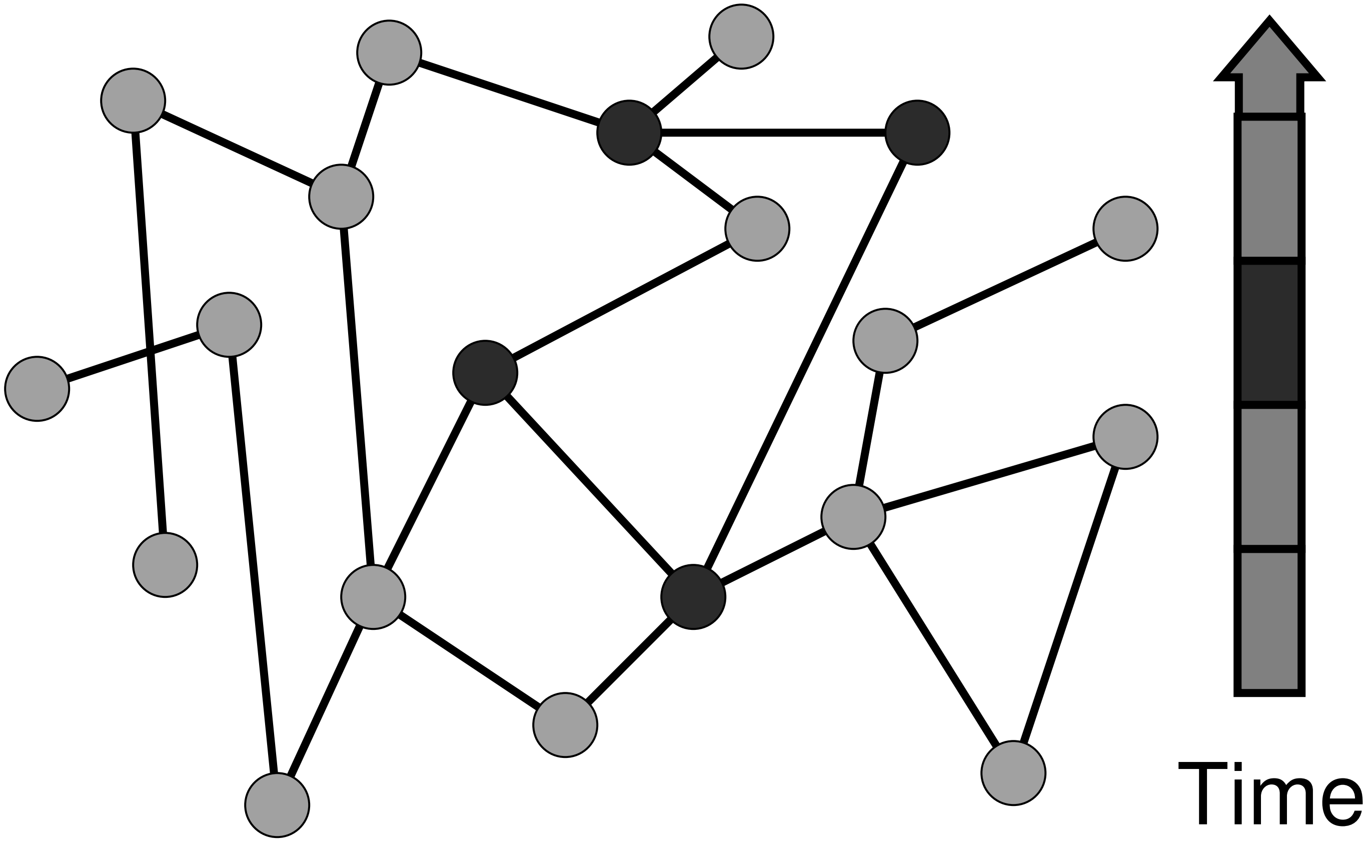
A Hidden Group



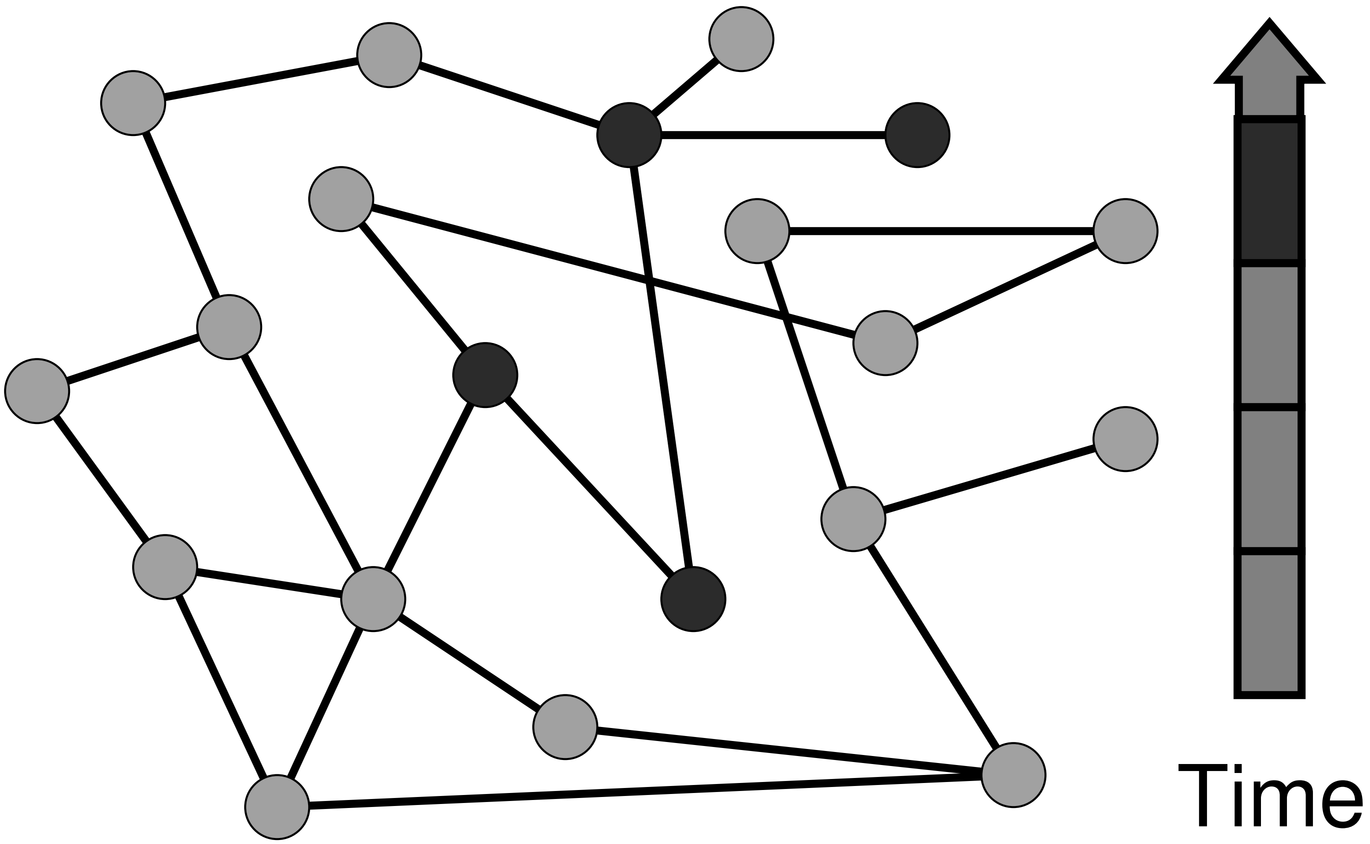
A Hidden Group



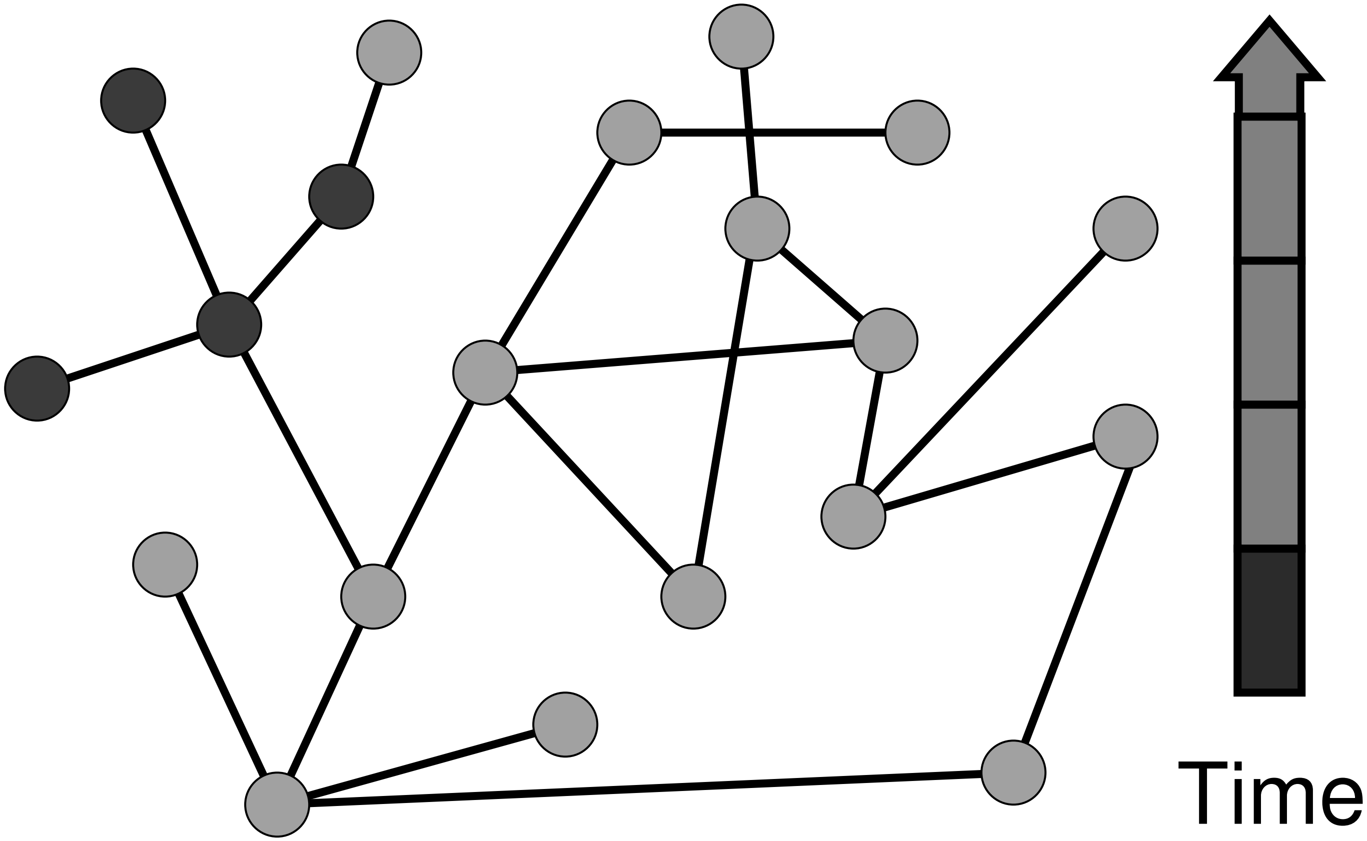
A Hidden Group



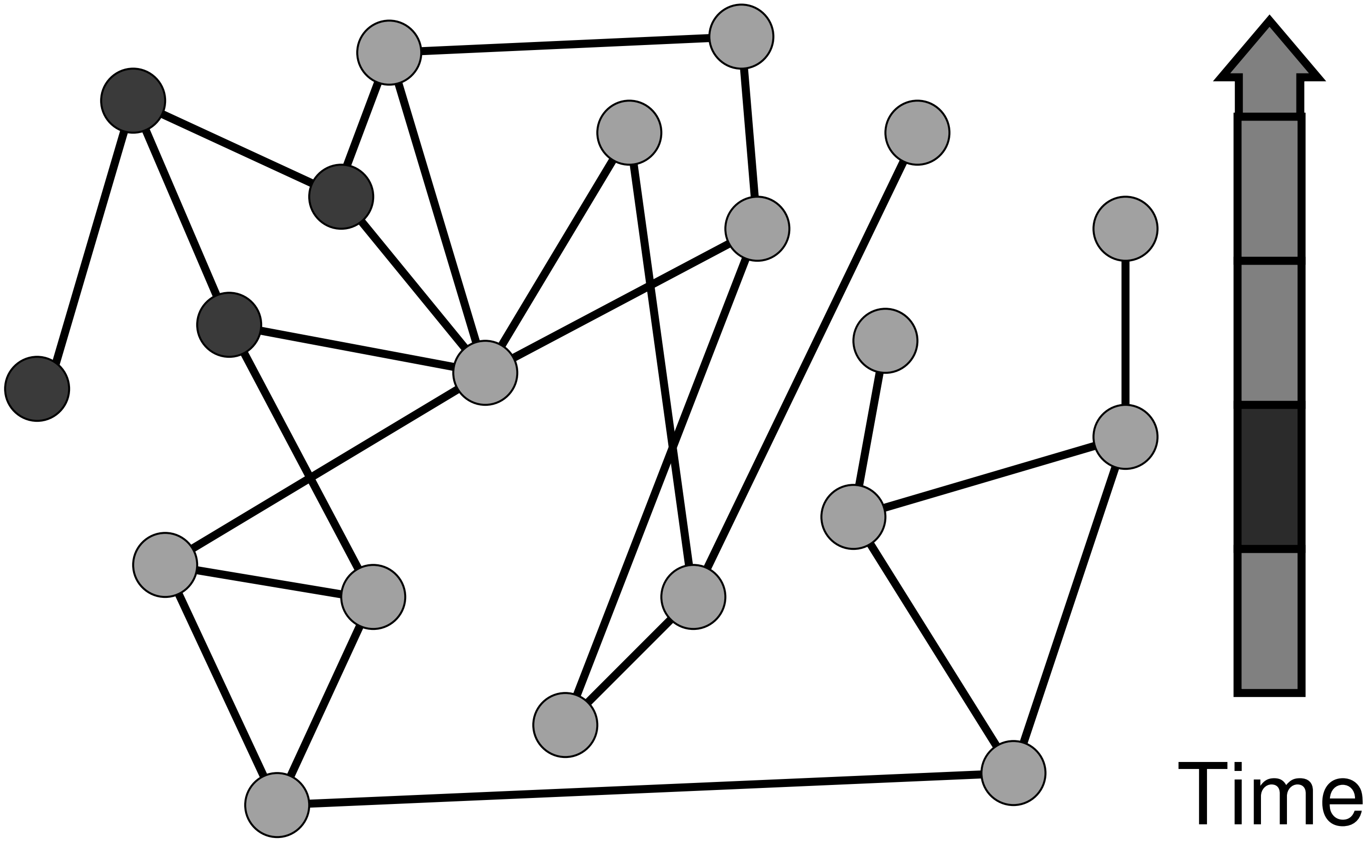
A Hidden Group



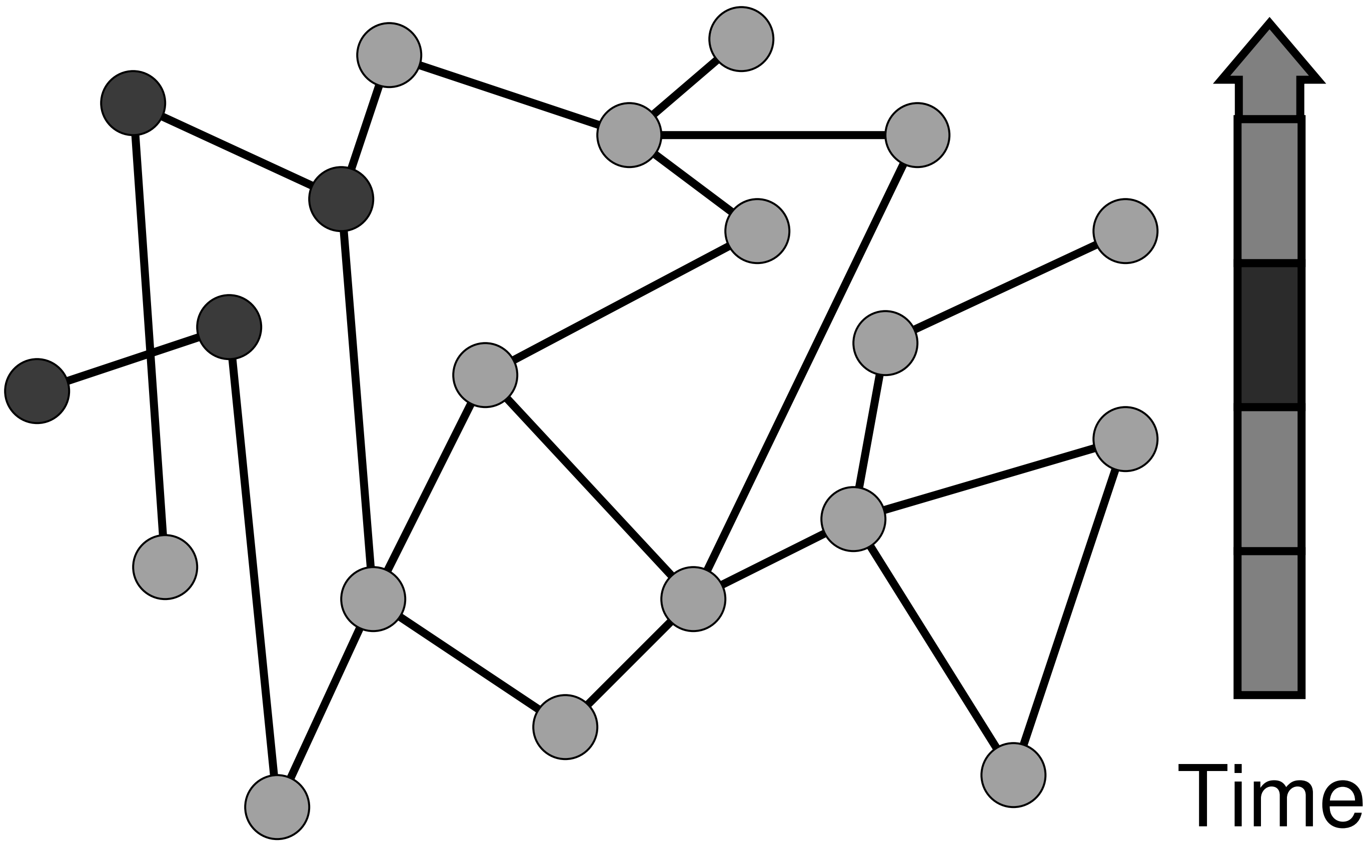
Not a Hidden Group



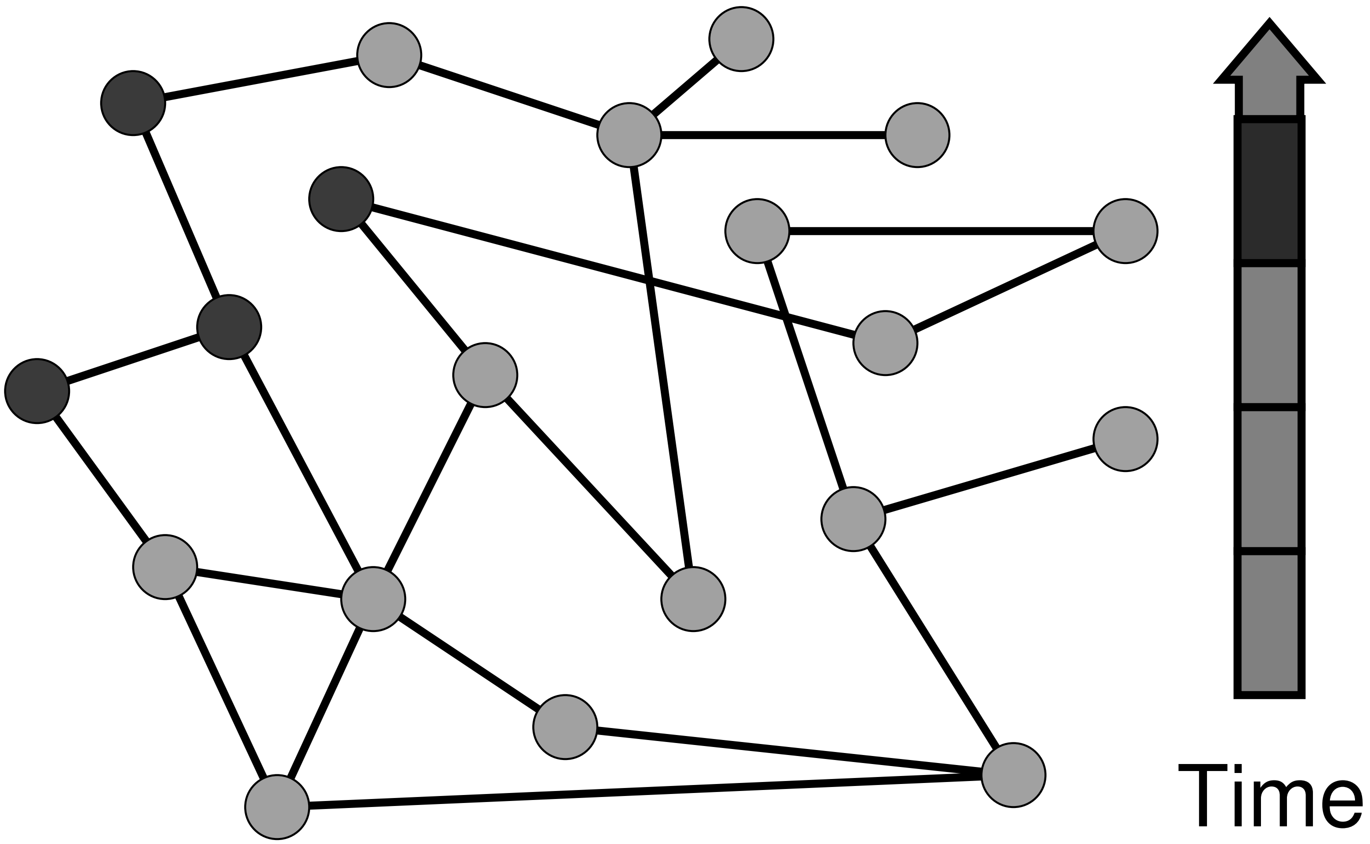
Not a Hidden Group



Not a Hidden Group



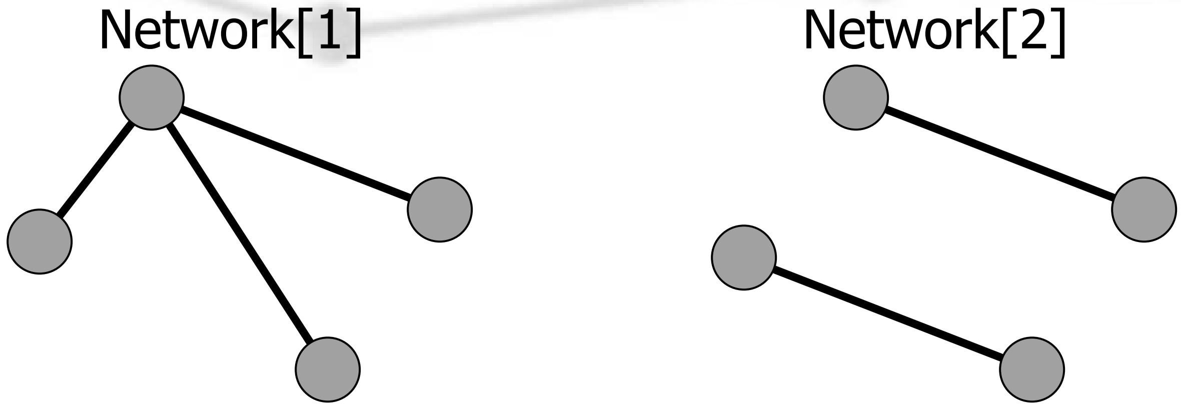
Not a Hidden Group



Overview

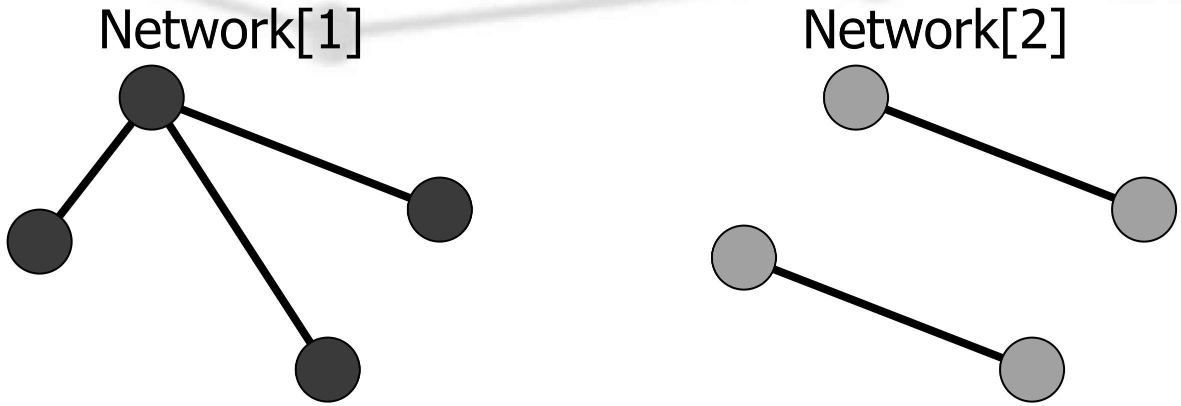
- Hidden group communication patterns
- Efficient discovery algorithm
- Background communication models
- Simulation results
- Conclusions

Algorithm for Discovering **Externally** Connected Groups



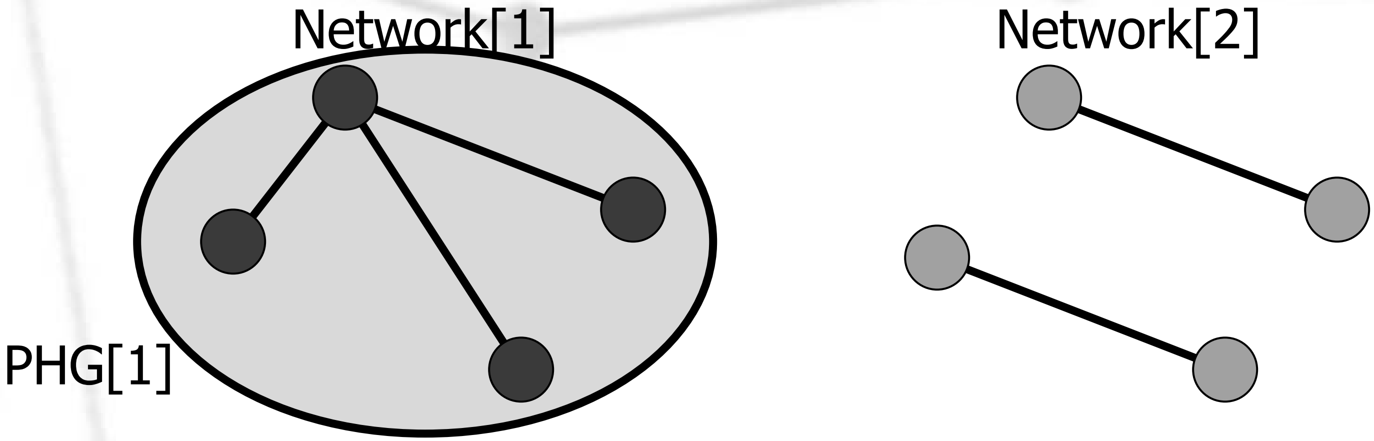
```
Find connected components of Network[1]
These components are PHG[1] (possible hidden groups)
For every remaining time step t :
    Find connected components of Network[t]
    PHG[t] is components intersected with PHG[t-1]
```

Algorithm for Discovering **Externally** Connected Groups



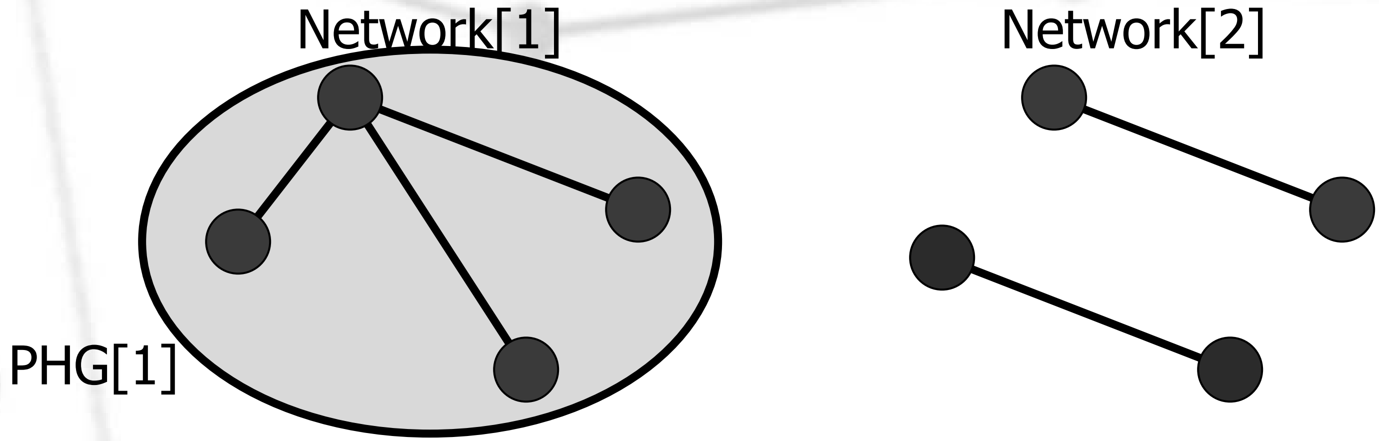
```
Find connected components of Network[1]
These components are PHG[1] (possible hidden groups)
For every remaining time step t :
    Find connected components of Network[t]
    PHG[t] is components intersected with PHG[t-1]
```

Algorithm for Discovering **Externally** Connected Groups



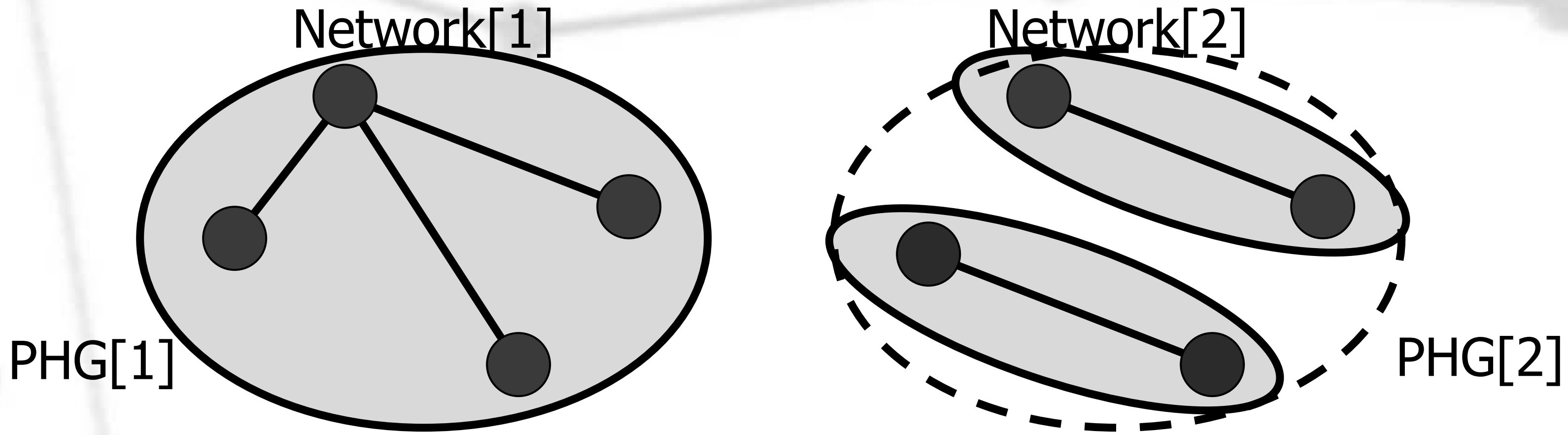
```
Find connected components of Network[1]
These components are PHG[1] (possible hidden groups)
For every remaining time step t :
    Find connected components of Network[t]
    PHG[t] is components intersected with PHG[t-1]
```

Algorithm for Discovering **Externally** Connected Groups



```
Find connected components of Network[1]
These components are PHG[1] (possible hidden groups)
For every remaining time step t :
    Find connected components of Network[t]
    PHG[t] is components intersected with PHG[t-1]
```

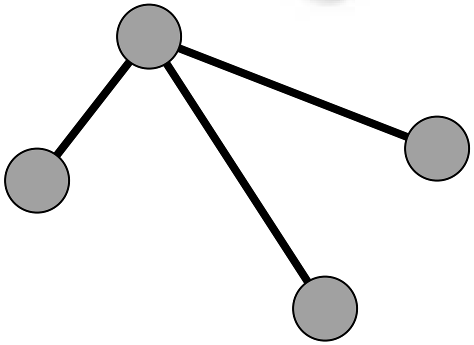
Algorithm for Discovering **Externally** Connected Groups



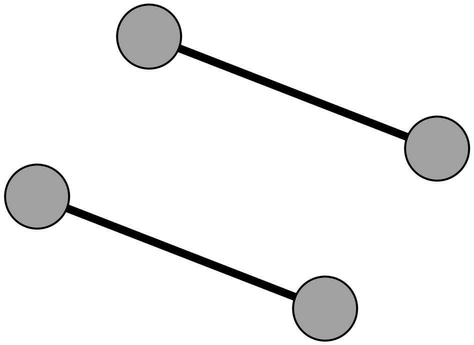
```
Find connected components of Network[1]
These components are PHG[1] (possible hidden groups)
For every remaining time step t :
    Find connected components of Network[t]
    PHG[t] is components intersected with PHG[t-1]
```

Algorithm for Discovering **Internally** Connected Groups

Network[1]

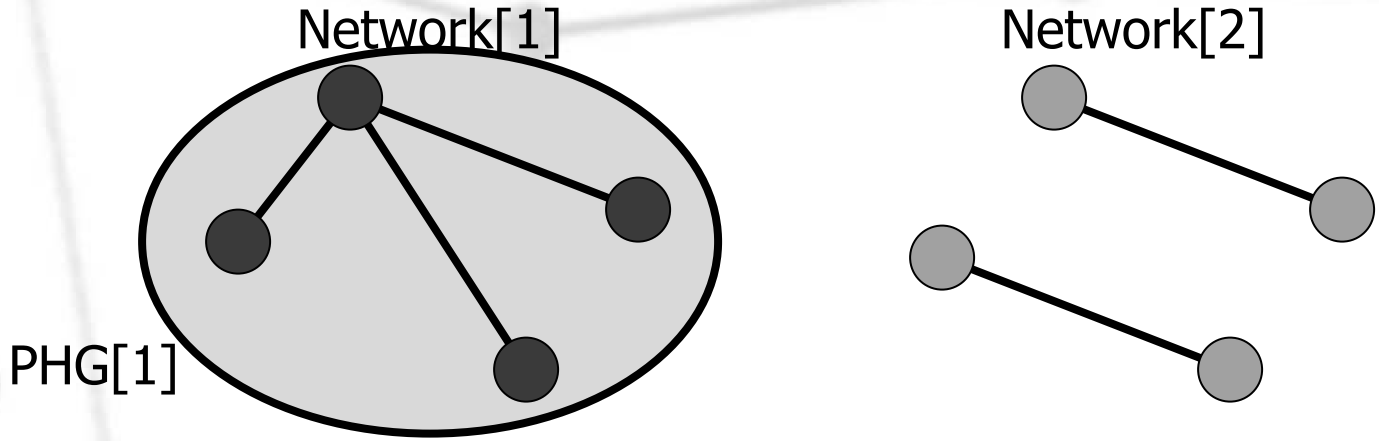


Network[2]



```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in all other networks
```

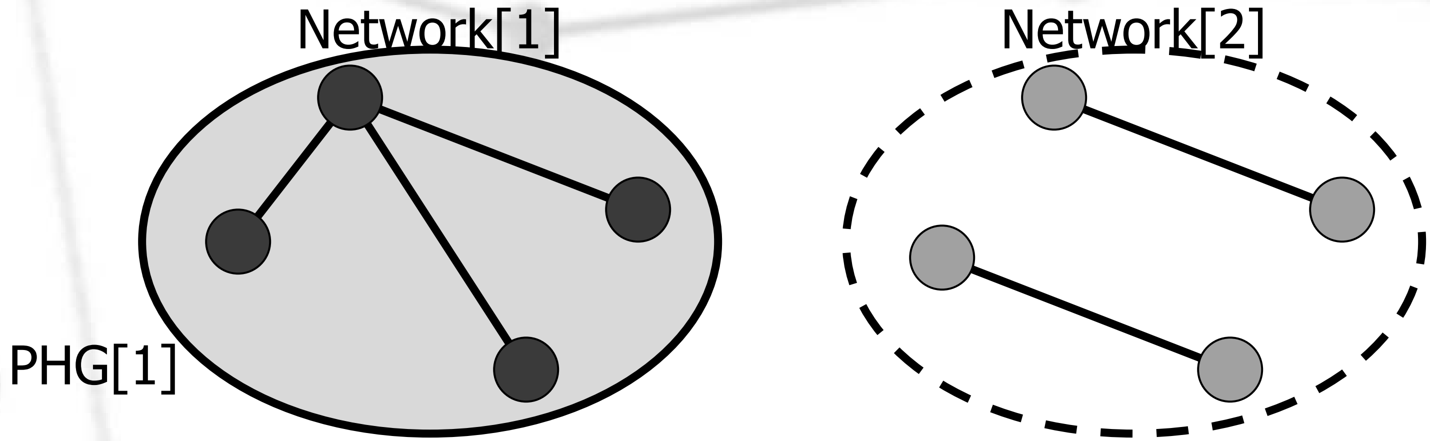
Algorithm for Discovering **Internally** Connected Groups



```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in all other networks
```

Rensselaer

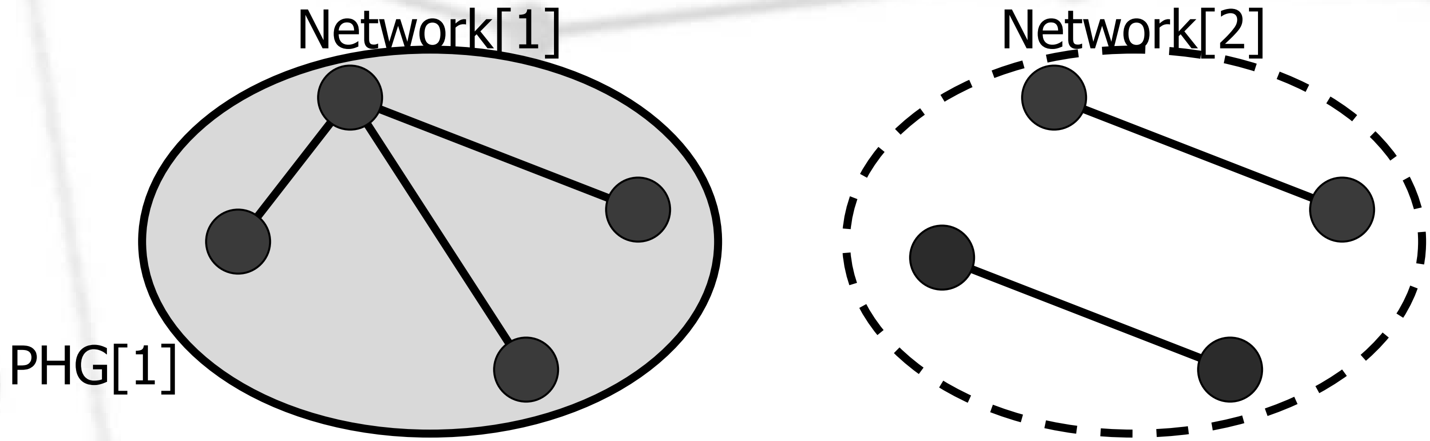
Algorithm for Discovering **Internally** Connected Groups



```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in all other networks
```

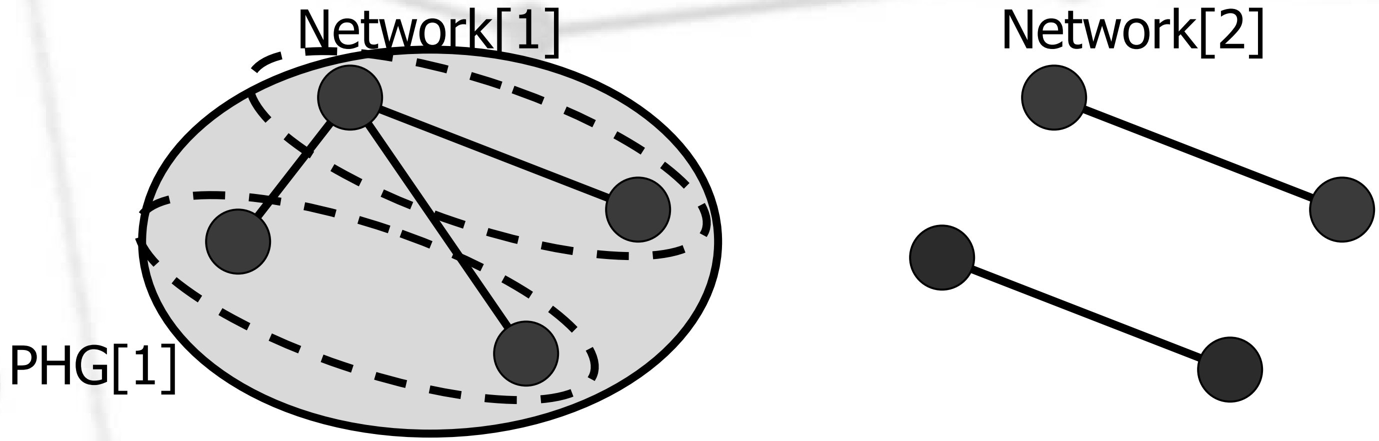
Rensselaer

Algorithm for Discovering **Internally** Connected Groups



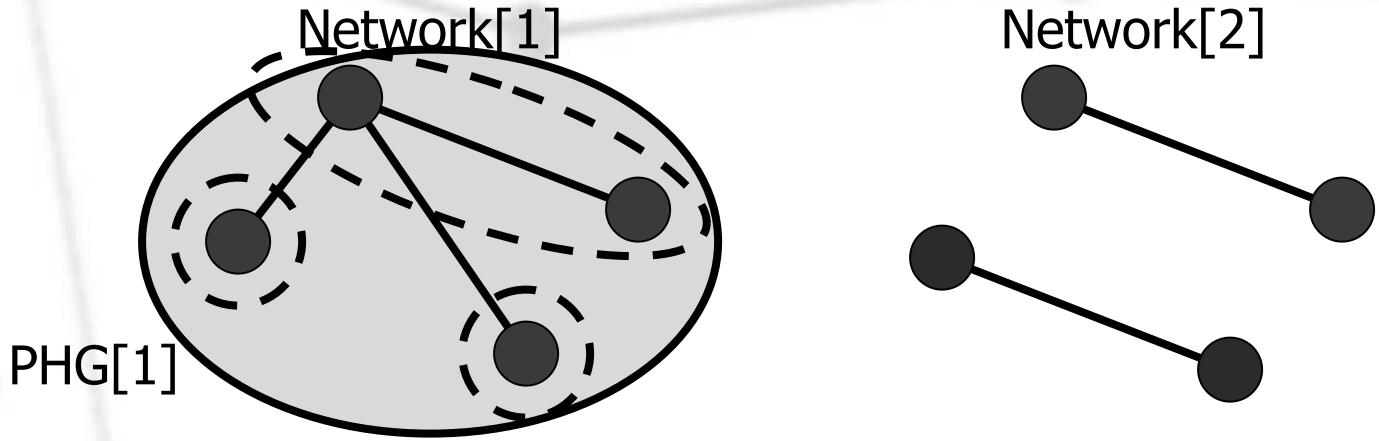
```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in all other networks
```

Algorithm for Discovering **Internally** Connected Groups



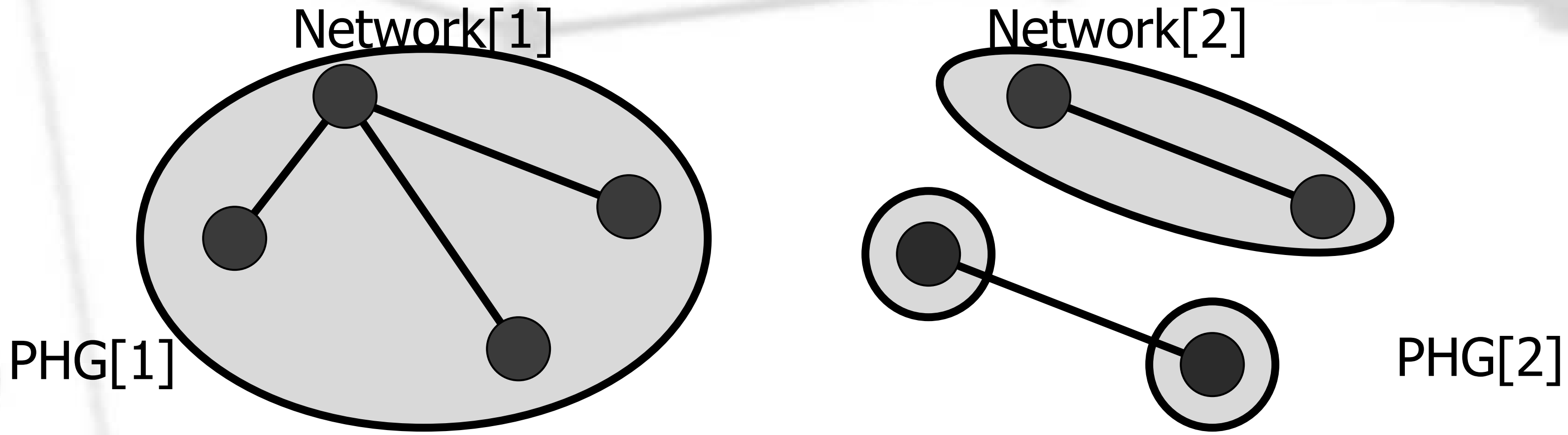
```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in all other networks
```

Algorithm for Discovering **Internally** Connected Groups



```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in all other networks
```

Algorithm for Discovering **Internally** Connected Groups



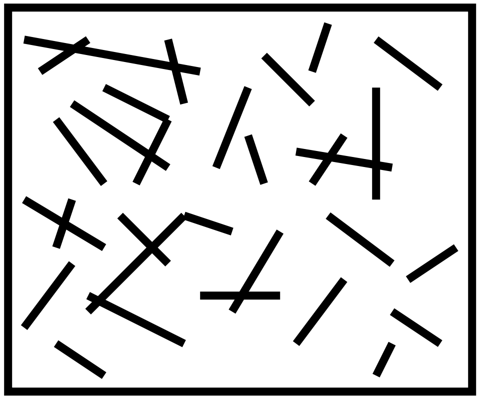
```
Find connected components of Network[1]
These components are PHG[1]
For every remaining time step t :
    For all groups in PHG[t-1] :
        If internally connected in Network[t], put in PHG[t]
        Otherwise break into components,
            check each component in *all* other networks
```

Overview

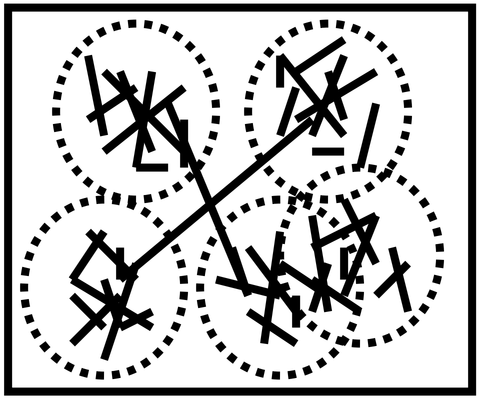
- Hidden group communication patterns
- Efficient discovery algorithm
- Background communication models
- Simulation results
- Conclusions

Background Communication Models

Uniform Random Graphs:
($G(n,p)$ Graphs)
Links spread uniformly



Group Random Graphs:
Most communication
occurs within groups



Overview

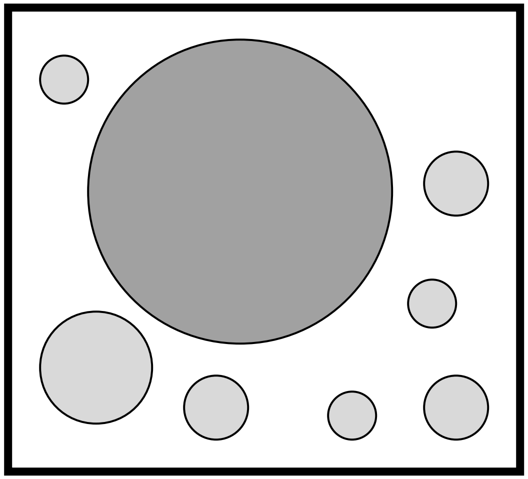
- Hidden group communication patterns
- Efficient discovery algorithm
- Background communication models
- Simulation results
- Conclusions

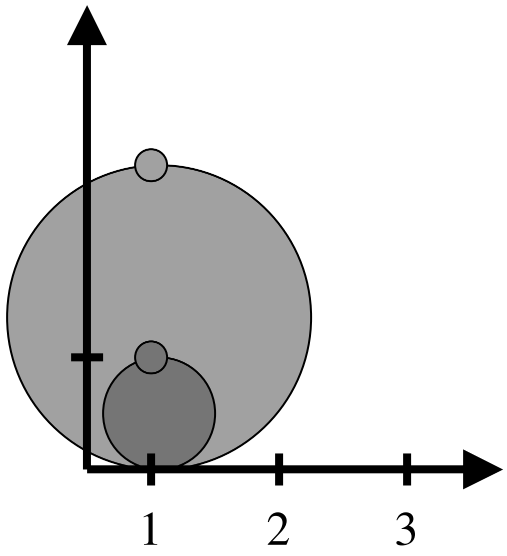
Discovery Time

- How much data is needed?
- Given a hidden group size h :
  - How long until the hidden group is discovered? $T(h)$
  - Under what conditions are hidden groups discovered quickly?

Discovery Time

PHG[1]

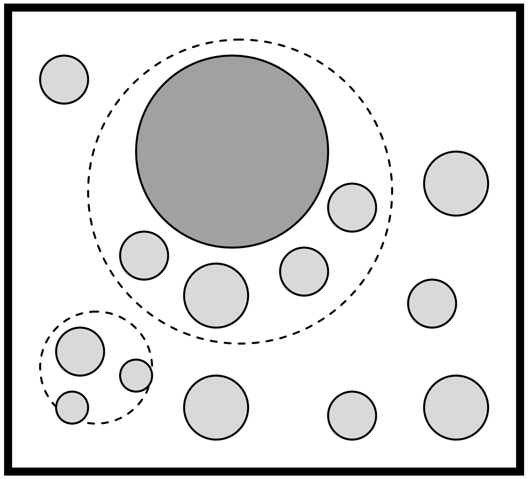


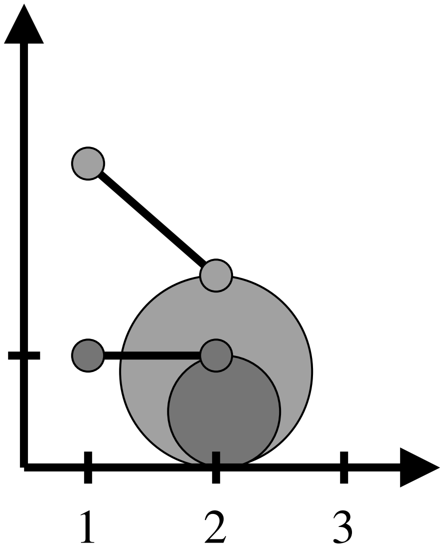


Hidden group size *h* : ●

Discovery Time

PHG[2]

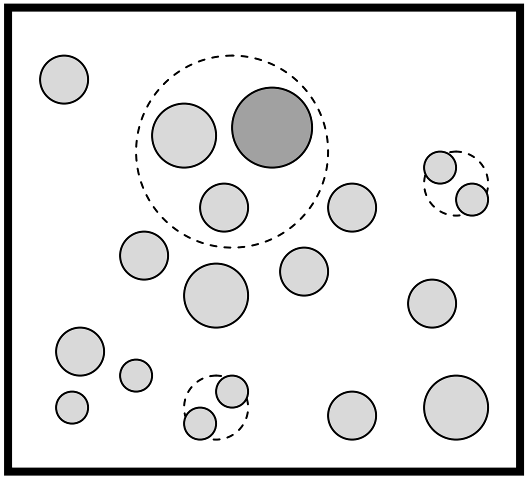


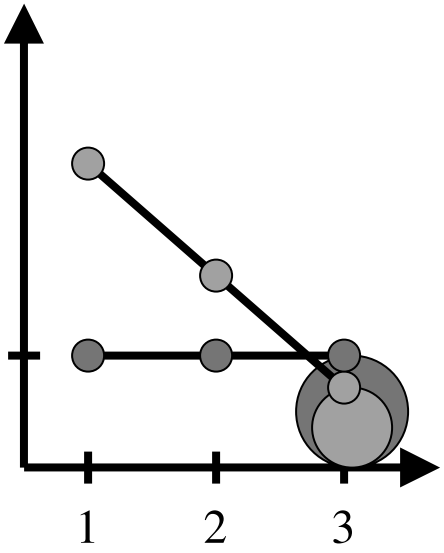


Hidden group size *h* : ●

Discovery Time

PHG[3]





Hidden group size h : ●

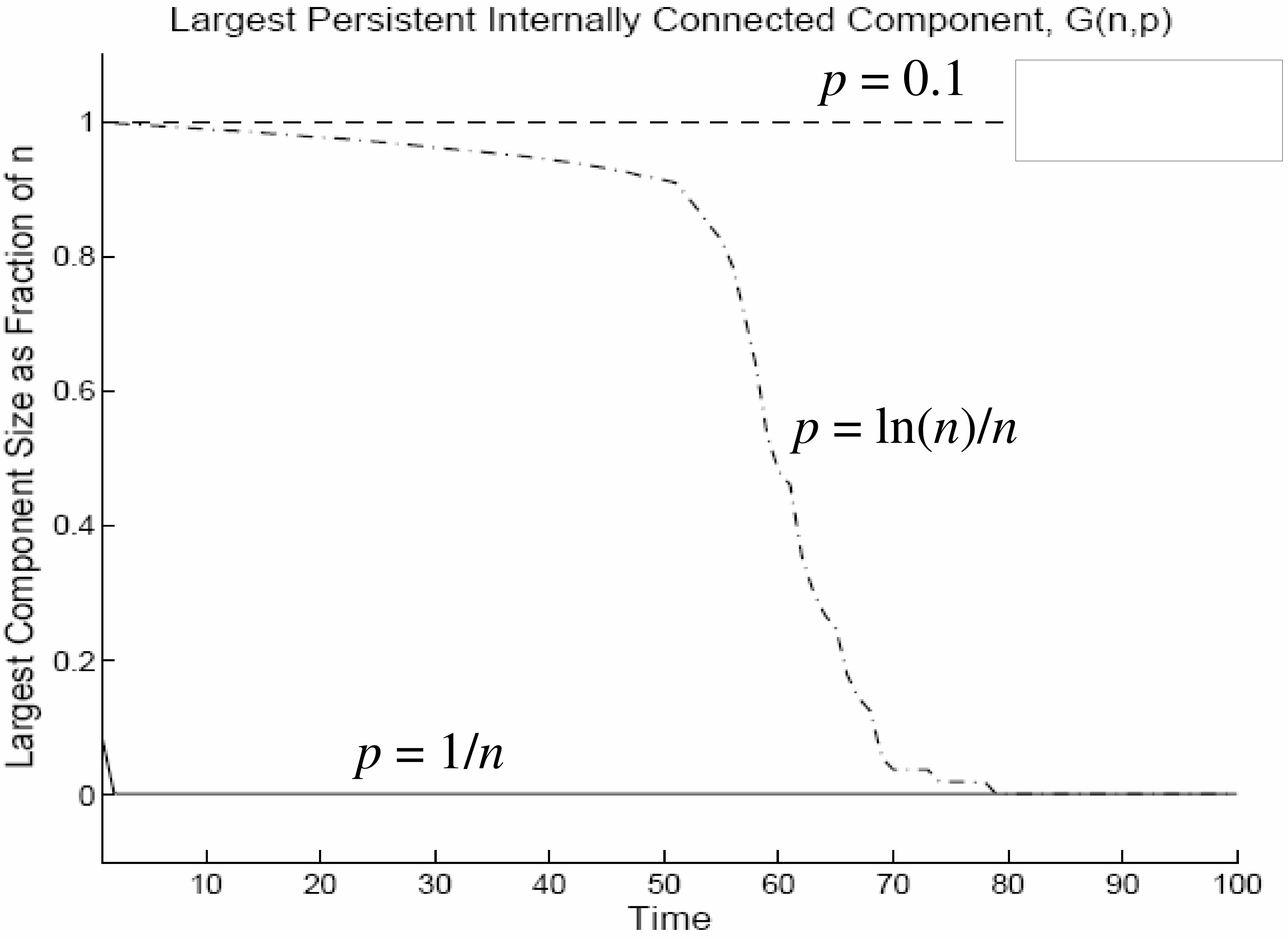
Theoretical *G*(*n*,*p*) Results

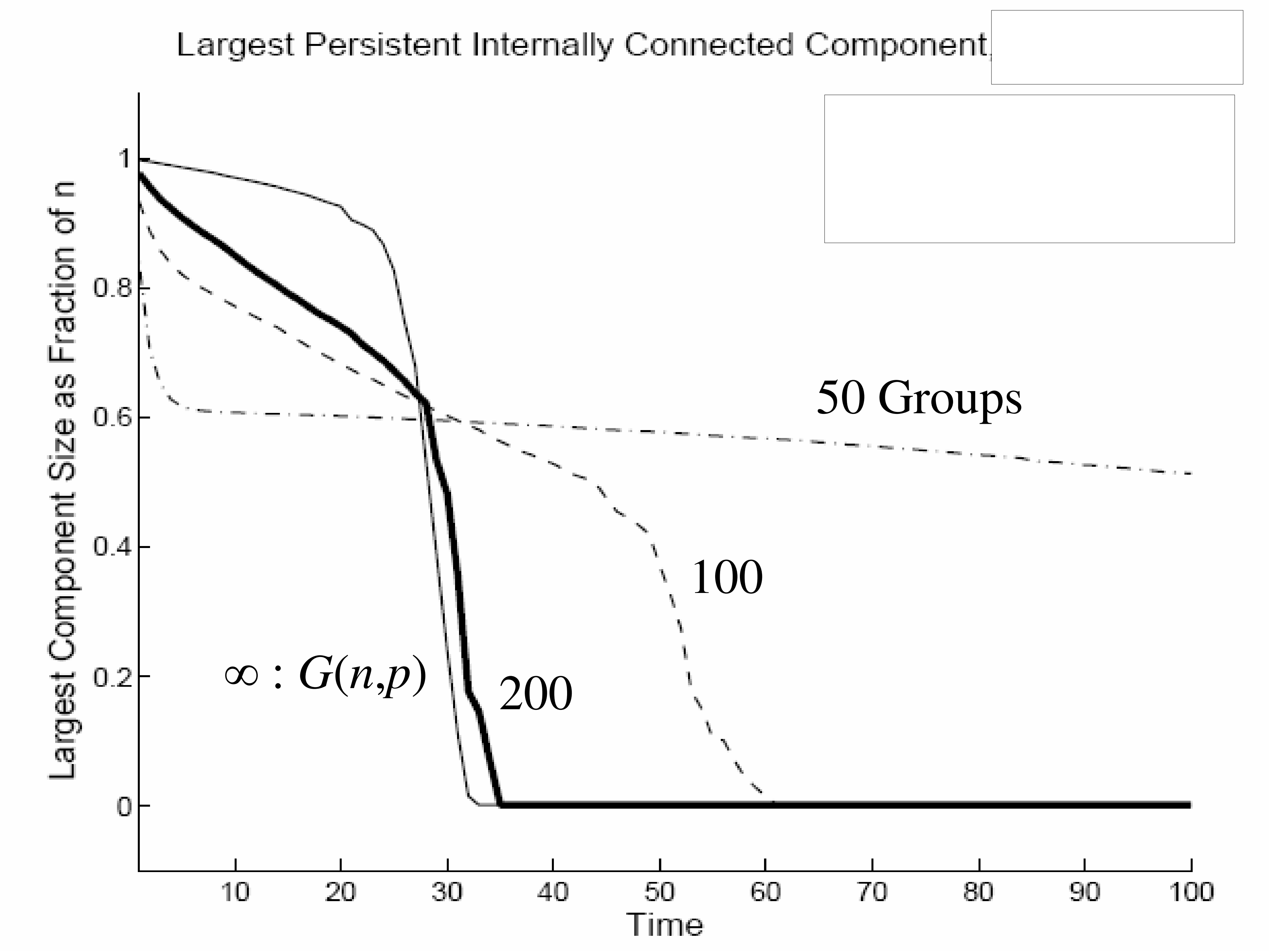
Largest connected subgraph:

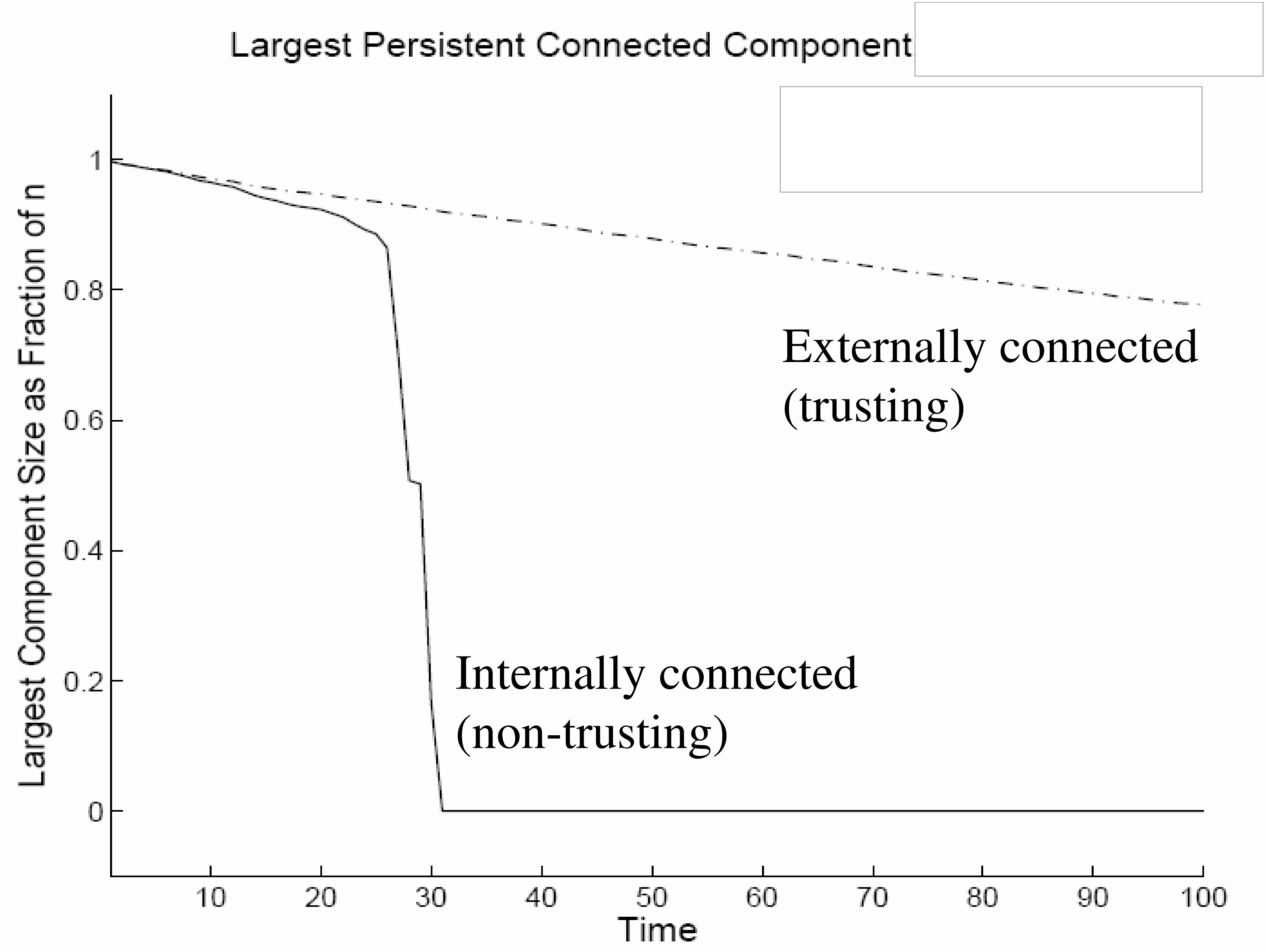
$$p = \frac{c}{n} \rightarrow \begin{array}{ll} O(\ln n) & 0 < c < 1 \\ O(n^{2/3}) & c = 1 \\ \beta(c)n & c > 1, \beta(c) < 1 \end{array}$$

$$p = \frac{\ln n}{n} + \frac{x}{n},\ x > 0$$

$$\rightarrow\ n \text{ with prob. } e^{-e^{-x}}$$


Largest Persistent Internally Connected Component, G(n,p)
p = 0.1
p = ln(n)/n
p = 1/n
Largest Component Size as Fraction of n
Time
0
0.2
0.4
0.6
0.8
1
10
20
30
40
50
60
70
80
90
100


Largest Persistent Internally Connected Component
Largest Component Size as Fraction of n
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
60
70
80
90
100
Time
50 Groups
100
200
∞ : $G(n,p)$


Largest Persistent Connected Component
Largest Component Size as Fraction of n
Time
0
0.2
0.4
0.6
0.8
1
10
20
30
40
50
60
70
80
90
100
Externally connected
(trusting)
Internally connected
(non-trusting)

Overview

- Hidden group communication patterns
- Efficient discovery algorithm
- Background communication models
- Simulation results
- Conclusions

Conclusions

When is it easier to discover hidden groups:

- Less intense background
- Less structured background
- Non-trusting hidden groups

Future Work

- Generalize hidden group pattern

NP-hard

- Evolving background groups
- Practical approaches
  - Some actors are flagged
  - More structured internal hidden group communications