Maintenance of random logical networks

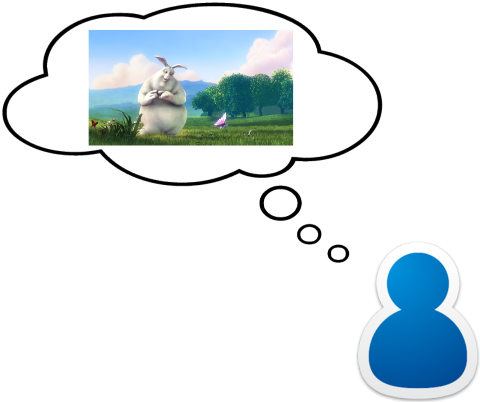
Romaric Duvignau

DCS seminar, CHALMERS

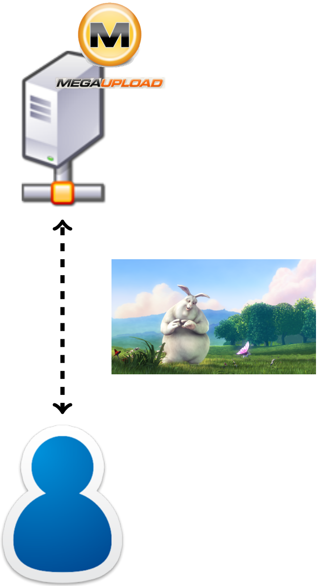
October 4, 2017

Plan

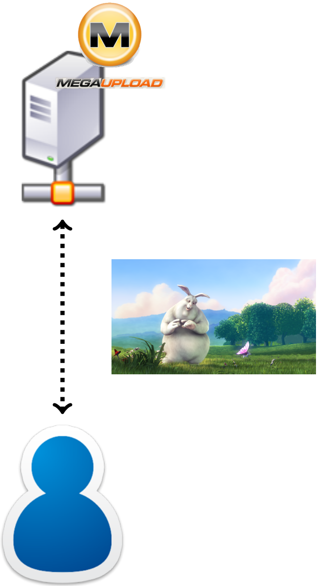
1. **A Quick Example of P2P Networks**
2. A New Model of Evolution
3. A Concrete Example: Uniform k-out Random Graphs
4. A More General Question



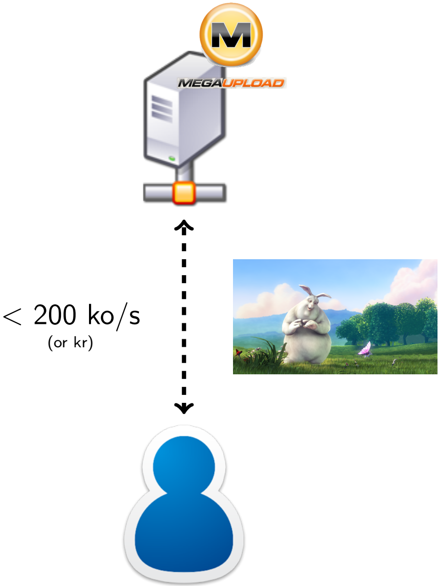
Motivation: modelling of P2P networks 1/2



Motivation: modelling of P2P networks 1/2



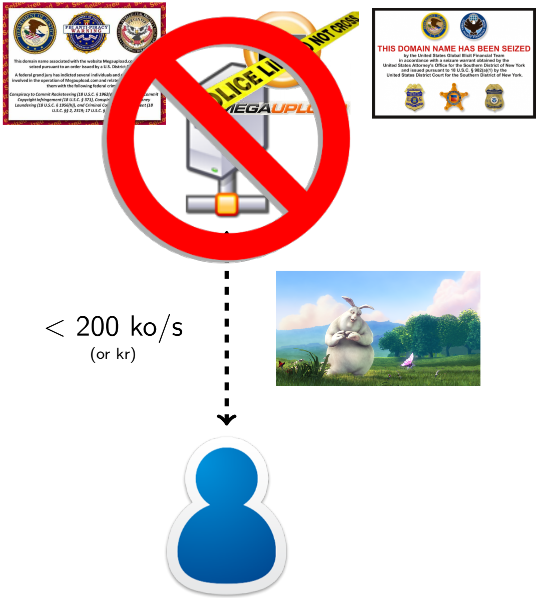
Motivation: modelling of P2P networks 1/2



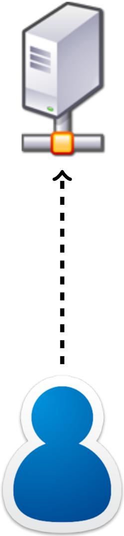
Motivation: modelling of P2P networks 1/2

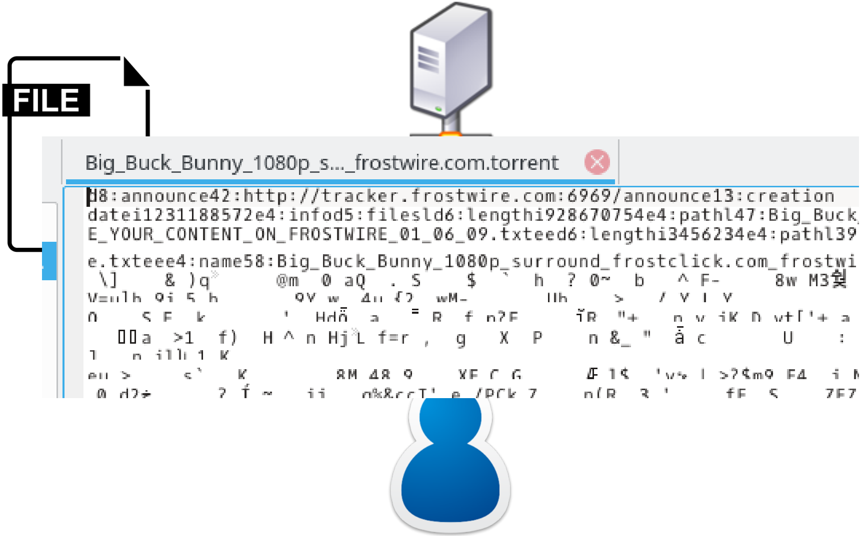
< 200 ko/s
(or kr)

Motivation: modelling of P2P networks 1/2

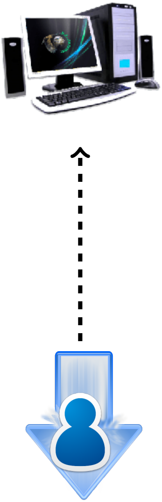


Motivation: modelling of P2P networks 2/2

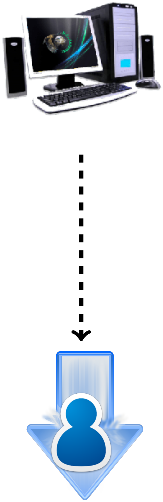



FILE
Big_Buck_Bunny_1080p_s..._frostwire.com.torrent
d8:announce42:http://tracker.frostwire.com:6969/announce13:creation
datei1231188572e4:infod5:filesld6:lengthi928670754e4:pathl47:Big_Buck
E_YOUR_CONTENT_ON_FROSTWIRE_01_06_09.txteed6:lengthi3456234e4:pathl39
e.txteee4:name58:Big_Buck_Bunny_1080p_surround_frostclick.com_frostwi

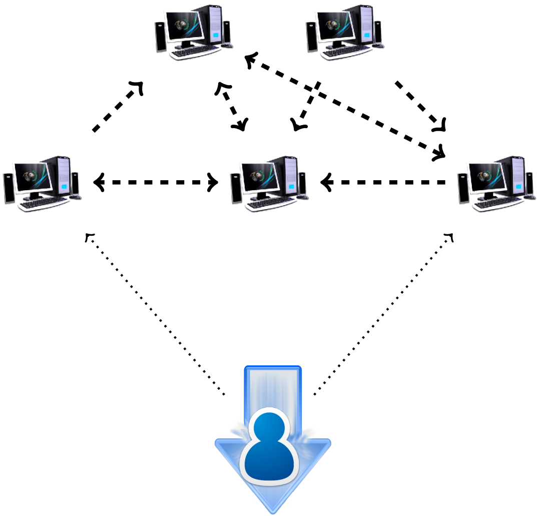
Motivation: modelling of P2P networks 2/2



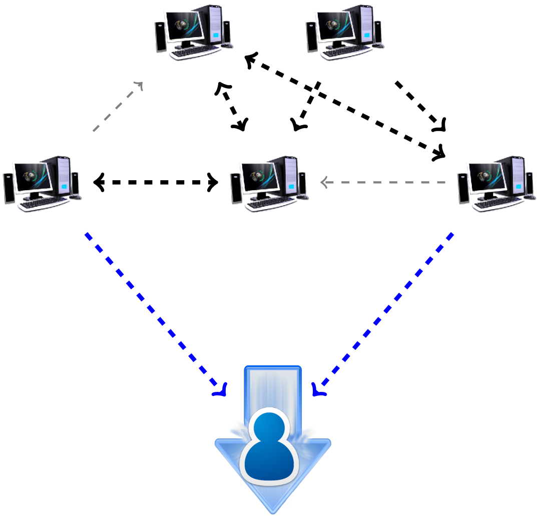
Motivation: modelling of P2P networks 2/2



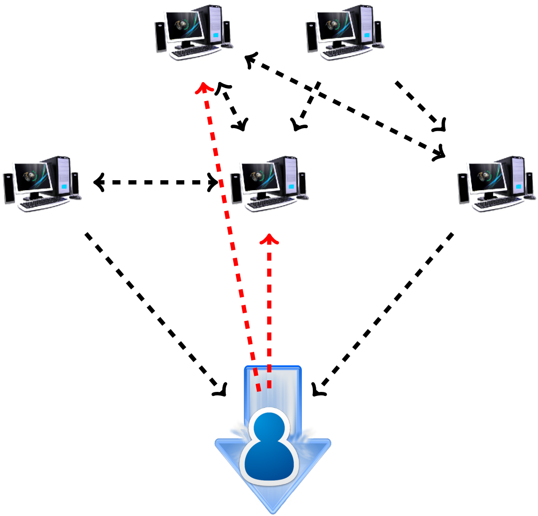
Motivation: modelling of P2P networks 2/2



Motivation: modelling of P2P networks 2/2



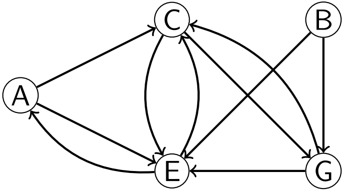
Motivation: modelling of P2P networks 2/2



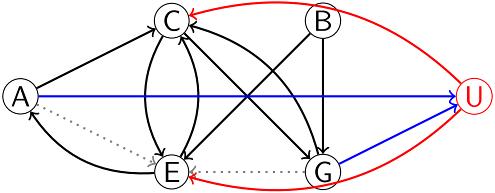
Plan

1. A Quick Example of P2P Networks
2. **A New Model of Evolution**
3. A Concrete Example: Uniform k-out Random Graphs
4. A More General Question

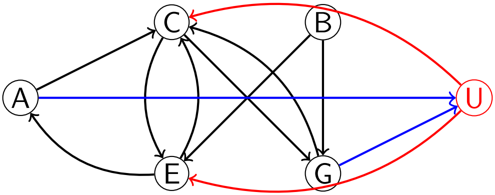
Introduction: Logical, Decentralized, Dynamic networks



Introduction: Logical, Decentralized, Dynamic networks



Introduction: Logical, Decentralized, Dynamic networks



Why should we look for good models ?

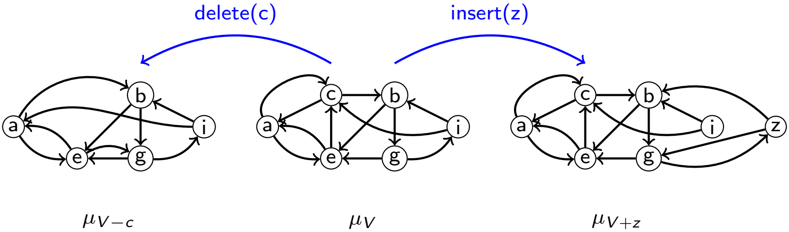
- analysis of the evolution of some concrete networks
- analysis of distributed algorithms running over such networks
- simulations of distributed algorithms operating over such networks (including adaptive algorithms working over dynamic networks)

Our model of evolution

Constrain the evolution to always stick to the target model

Our model of evolution

Constrain the evolution to always stick to the target model



Context : evolution of $\mathrm{P2P}$ networks

- **In the literature**, some *good properties* of the network are maintained over time, but implies :
  - difficulties in update alogrithms' conception leading to complex procedures
  - dynamicity modelled by a probabilistic process (Poisson) : **non realistic** [Pouwelse *et al*, 2005]
  - difficult analysis without those hypothesis (analysis under simplistic update models : insertion only, fifo)

Typical good properties

- small degree, small diameter, high connectivity, etc

Context : evolution of $\mathrm{P2P}$ networks

- **In the literature**, some *good properties* of the network are maintained over time, but implies :
  - difficulties in update alogrithms' conception
  - dynamicity modelled by a probabilistic process (Poisson) : **non realistic**
  - difficult analysis without those hypothesis

Typical good properties

- small degree, small diameter, high connectivity, etc

Context : evolution of P2P networks

- **In the literature**, some *good properties* of the network are maintained over time, but implies :
  - difficulties in update alogrithms' conception
  - dynamicity modelled by a probabilistic process (Poisson) : **non realistic**
  - difficult analysis without those hypothesis

- **Our solution** : *randomness preservation*

Typical good properties

- small degree, small diameter, high connectivity, etc

Context : evolution of P2P networks

- **In the literature**, some *good properties* of the network are maintained over time, but implies :
  - difficulties in update alogrithms' conception
  - dynamicity modelled by a probabilistic process (Poisson) : **non realistic**
  - difficult analysis without those hypothesis

- **Our solution** : *randomness preservation*, answers those problems:
  - properties are always maintained
  - analysis is simplified
  - it is not influenced by an **adversarial** sequence of updates
  - no drift phenomena

Typical good properties

- small degree, small diameter, high connectivity, etc

Local update algorithms

- $\mathcal{LOCAL}$ model (synchronous, error-free, message passing)
- Two submodels: exact size of the network **known** or **unknown** to the participating nodes

Introduction: An Optimistic First Contact Model

Local update algorithms

- $\mathcal{LOCAL}$ model (synchronous, error-free, message passing)
- Two submodels: exact size of the network **known** or **unknown** to the participating nodes
- **"First Contact":** Every node has access to a global primitive which samples uniformly a node over the entire network

$$\mathit{RandomVertex}()$$

Introduction: An Optimistic First Contact Model

Local update algorithms

- $\mathcal{LOCAL}$ model (synchronous, error-free, message passing)
- Two submodels: exact size of the network **known** or **unknown** to the participating nodes
- **"First Contact":** Every node has access to a global primitive which samples uniformly a node over the entire network

$$\mathit{RandomVertex}()$$

An optimistic first contact but...

- Optimistic: uniformity, reusability, availability

Introduction: An Optimistic First Contact Model

Local update algorithms

- $\mathcal{LOCAL}$ model (synchronous, error-free, message passing)
- Two submodels: exact size of the network **known** or **unknown** to the participating nodes
- **"First Contact":** Every node has access to a global primitive which samples uniformly a node over the entire network

$$\mathit{RandomVertex}()$$

An optimistic first contact but...

- Optimistic: uniformity, reusability, availability
- May be approximated in practice: uniform hash (CHORD), centralized cache, random walks, dissemination of tokens, etc

Introduction: An Optimistic First Contact Model

Local update algorithms

- $\mathcal{LOCAL}$ model (synchronous, error-free, message passing)
- Two submodels: exact size of the network **known** or **unknown** to the participating nodes
- **"First Contact":** Every node has access to a global primitive which samples uniformly a node over the entire network

$$\mathit{RandomVertex}()$$

An optimistic first contact but...

- Optimistic: uniformity, reusability, availability
- May be approximated in practice: uniform hash (Chord), centralized cache, random walks, dissemination of tokens, etc

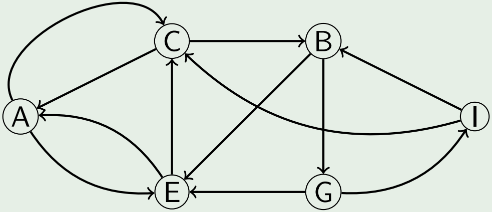
What about the cost model ?

Plan

1. A Quick Example of P2P Networks
2. A New Model of Evolution
3. **A Concrete Example: Uniform k-out Random Graphs**
4. A More General Question

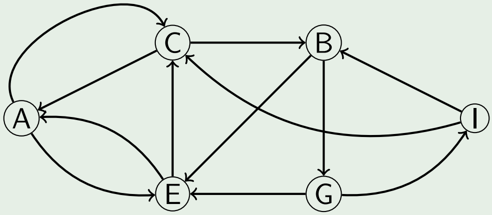
Uniform k-out graphs

Example of a 2-out graph



Uniform k-out graphs

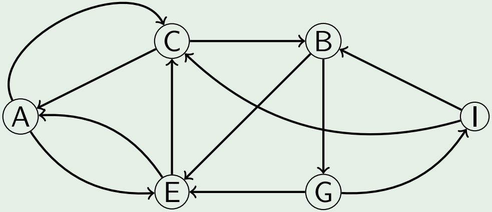
Example of a 2-out graph



- Directed graphs with no loops and where each vertex has exactly k out-neighbours

Uniform k-out graphs

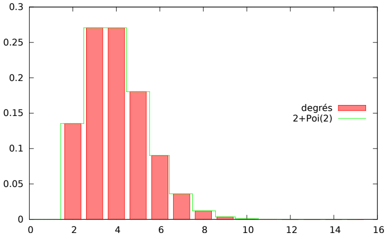
Example of a 2-out graph



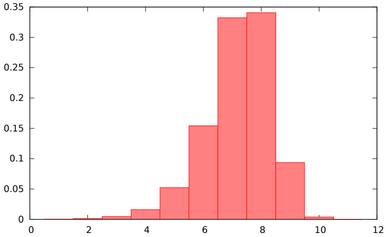
- Directed graphs with no loops and where each vertex has exactly k out-neighbours
- The uniform distribution over vertex set V is equivalent to:
  - For each $v \in V$, the outgoing neighbourhood of v is a uniform k-subset of $V - v$
  - All outgoing neighbourhood are independent

Why uniform k-out graphs ?

Figure : Some statistics of 2-out random graphs.

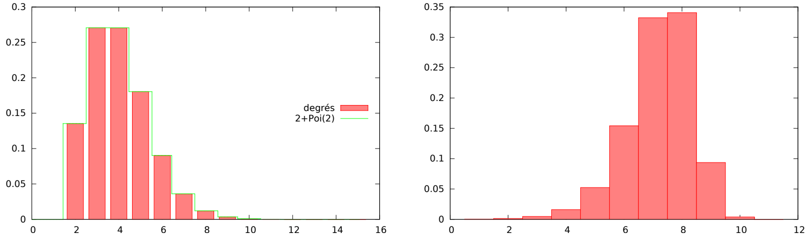


(a) Degrees distribution for $G^2_{10^7}$



(b) Distances distribution for $G^2_{10^4}$

Figure : Some statistics of 2-out random graphs.

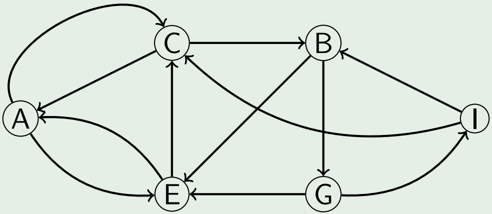


(a) Degrees distribution for $G^2_{10^7}$ (b) Distances distribution for $G^2_{10^4}$

The uniform distribution is associated with good properties similar to those sought for $\mathrm{P2P}$ networks (small degree/diameter, high connectivity)

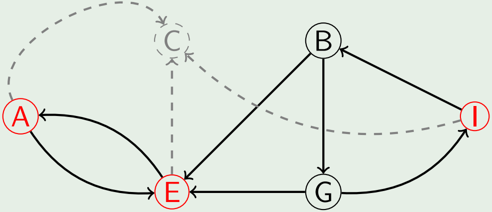
Maintenance of k-out graphs : deletion 1/2

Deletion of node C



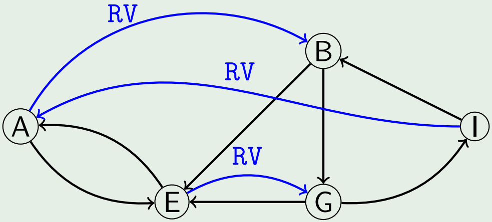
Node C wishes to leave the network.

Deletion of node C



Node C leaves the network, and 3 loosed edges are created.

Deletion of node C



Nodes A, E and I find a substitute node for node C using `RandomVertex()`.

In total, we need $k + \mathcal{O}(1/n)$ calls in average.

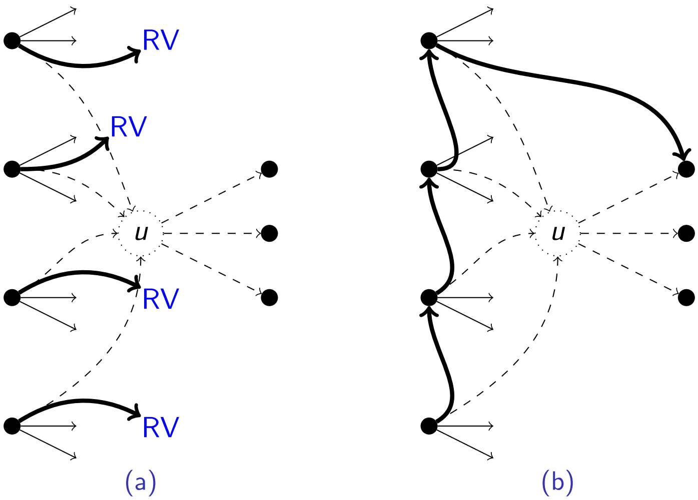
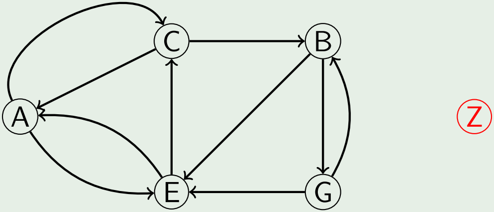


Figure : Typical deletion of node u in the two algorithms.

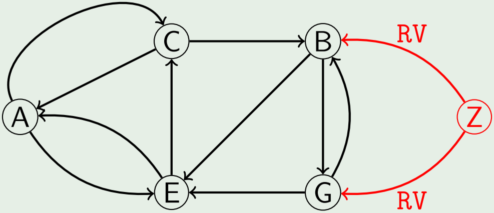
Maintenance of k-out graphs: insertion 1/2

Insertion of node Z



Node Z wishes to join the network.

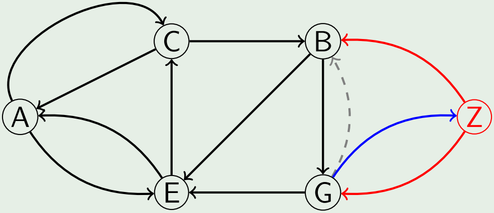
Insertion of node Z



Node Z chooses 2 distinct nodes as out-neighbours, using `RandomVertex()`.

In average, $k + \mathcal{O}(1/n)$ calls to the primitive are needed.

Insertion of node Z



Node Z chooses $X \sim \text{Binomial}(n,\ k/n)$ distinct nodes as in-neighbours, and *steal* one edge from each of them.

We need $k + \mathcal{O}(1/n)$ more calls in average to sample these vertices.

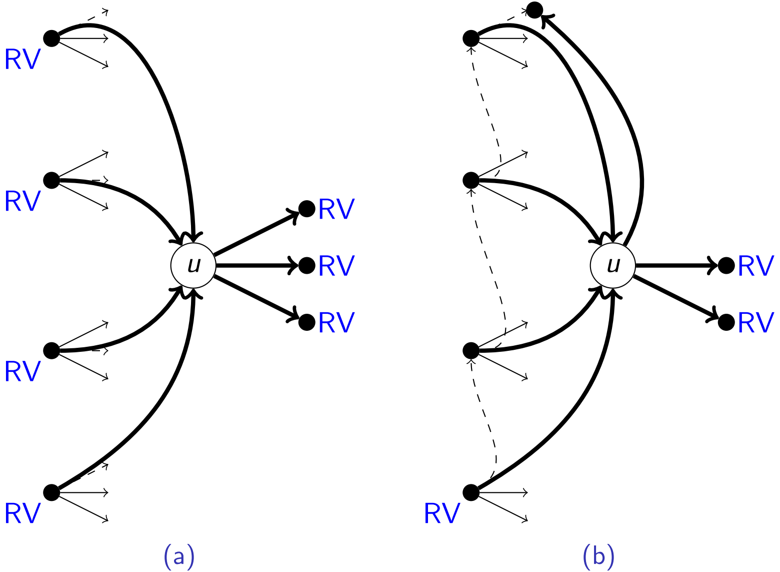


Figure : Typical instances of insertion of node u in the two algorithms.

Maintenance of k-out graphs

Theorem (Duchon, D., 14)

There exist local update algorithms in order to maintain uniform k-out graphs and:

- *modifying a minimal number of links,*
- *of average constant complexity, and using in average:*
  - $k + \mathcal{O}(1/n)$ *calls to* $\mathtt{RV}$ *for the insertion;*
  - $\mathcal{O}(1/n)$ *calls to* $\mathtt{RV}$ *for the deletion.*
- **and these bounds are asymptotically optimal.**

Maintenance of k-out graphs

Theorem (Duchon, D., 14)

There exist local update algorithms in order to maintain uniform k-out graphs and:

- *modifying a minimal number of links,*
- *of average constant complexity, and using in average:*
  - $k + \mathcal{O}(1/n)$ *calls to* `RV` *for the insertion;*
  - $\mathcal{O}(1/n)$ *calls to* `RV` *for the deletion.*
- **and these bounds are asymptotically optimal.**

All insertion procedures need to know the exact size of the network during update in order to simulate Binomial(n, k/n).

Maintenance of k-out graphs

Theorem (Duchon, D., 14)

There exist local update algorithms in order to maintain uniform k-out graphs and:

- *modifying a minimal number of links,*
- *of average constant complexity, and using in average:*
  - $k + \mathcal{O}(1/n)$ *calls to* $\texttt{RV}$ *for the insertion;*
  - $\mathcal{O}(1/n)$ *calls to* $\texttt{RV}$ *for the deletion.*
- **and these bounds are asymptotically optimal.**

All insertion procedures need to know the exact size of the network during update in order to simulate Binomial(n, k/n).

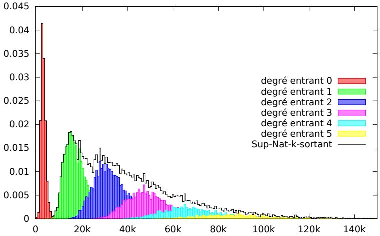
Simulation of Binomial(n, k/n) (D. 15)

This particular law can be simulated in our model with a non-trivial combinatorics-based algorithm using $< 5.2k + \mathcal{O}(1/n)$ expected calls to $\texttt{RV}$ without knowing n.

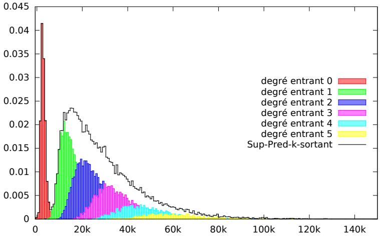
Algorithm for Binomial$(n, 1/n)$

```
m ← 1, g ← 0, j ← 1
S ← {Uniform(V)}
loop
    i ← Random(m + 1)
    if i = m + 1 then
        j ← j + 1
    else if i > g then
        j ← j − 1
        g ← m + 1
    else
        return j
    end if
    x ← Uniform(V)
    if x ∈ S then
        if g = m + 1 then
            return j + 1
        else
            return j − 1
        end if
    else
        S ← S + x
    end if
    m ← m + 1
end loop
```

Different algorithms have different consequences 1/2

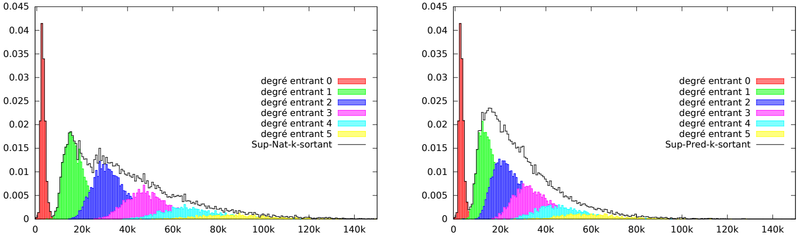


(a) SUP-NAT-K-SORTANT 10^4 nodes



(b) SUP-PRED-K-SORTANT 10^4 nodes

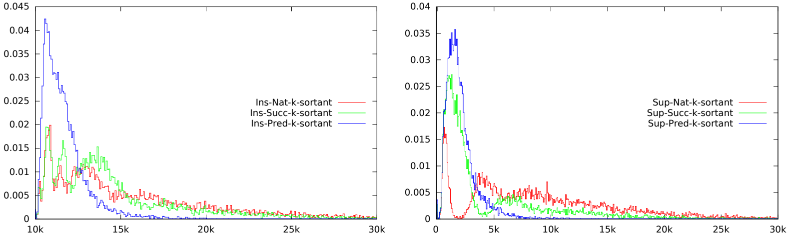
Figure : Number of modified distances after one deletion, $k = 2$.



(a) SUP-NAT-K-SORTANT 10^4 nodes (b) SUP-PRED-K-SORTANT 10^4 nodes

Figure : Number of modified distances after one deletion, $k = 2$.

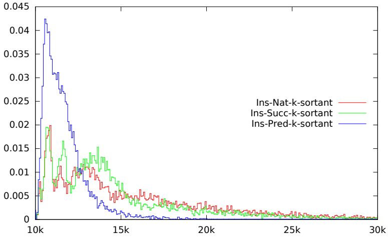
On these simulations, the second algorithm modifies 25% less distances in the graph.



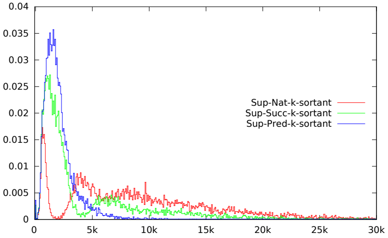
(a) Insertion 10^4 nodes (b) Deletion 10^4 nodes

Figure : Number of modified distances by more than one unit, $k = 2$.

Different algorithms have different consequences 2/2



(a) Insertion 10^4 nodes



(b) Deletion 10^4 nodes

Figure : Number of modified distances by more than one unit, $k = 2$.

On these simulations, the second algorithms modify respectively 25% less distances during insertion and about 80% less distances during deletion.

Plan

1. A Quick Example of P2P Networks
2. A New Model of Evolution
3. A Concrete Example: Uniform k-out Random Graphs
4. **A More General Question**

A general question as a starting point

Coul any **topology** of networks **arise**
and **last** in a **distributed** fashion ?

A general question as a starting point

random graphs

Coul any **topology** of networks **arise** and **last** in a **distributed** fashion ?

random graphs

build the network from scratch

Coul any **topology** of networks **arise**
and **last** in a **distributed** fashion ?

A general question as a starting point

random graphs

build the network from scratch

Coul any **topology** of networks **arise**
and **last** in a **distributed** fashion ?

nodes joining and leaving

A general question as a starting point

random graphs

build the network from scratch

Coul any **topology** of networks **arise**

and **last** in a **distributed** fashion ?

nodes joining and leaving

$\mathcal{LOCAL}$ model

(+ **first contact**)

A general question as a starting point

random graphs

build the network from scratch

Coul any **topology** of networks **arise**
and **last** in a **distributed** fashion ?

nodes joining and leaving

$\mathcal{LOCAL}$ model
(+ **first contact**)

And what about the cost ?

A general question as a starting point

random graphs

build the network from scratch

Coul any **topology** of networks **arise**

and **last** in a **distributed** fashion ?

nodes joining and leaving

$\mathcal{LOCAL}$ model

(+ **first contact**)

And what about the cost ?

Applications: model the evolution of P2P-like networks

Introduction: Networks modelled as Random Graphs

Some classical models of logical networks

- **Erdős–Rényi** $G(n, p)$:
  - p fixed: dense graphs, binomial degrees (*similar* degrees), ...
  - $p = p(n)$: phase transition around $\log(n)/n$, ...
- **Uniform** k**-regular graphs**:
  - constant degree, high connectivity (for $k \geq 3$), logarithmic diameter, ...
- **Barabási–Albert** (scale-free networks):
  - preferential attachment, power-law degree distribution, ...

Introduction: Networks modelled as Random Graphs

Some classical models of logical networks

- **Erdős–Rényi** $G(n, p)$:
  - p fixed: dense graphs, binomial degrees (*similar* degrees), ...
  - $p = p(n)$: phase transition around $\log(n)/n$, ...
- **Uniform** k**-regular graphs**:
  - constant degree, high connectivity (for $k \geq 3$), logarithmic diameter, ...
- **Barabási–Albert** (scale-free networks):
  - preferential attachment, power-law degree distribution, ...

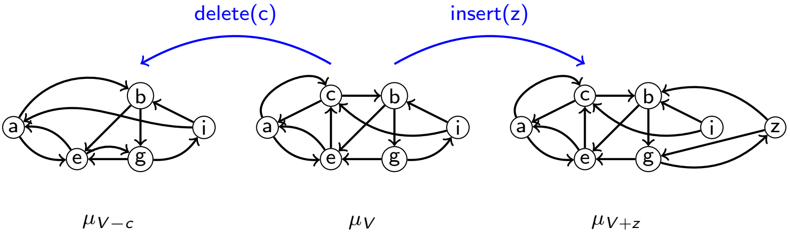
Could these classical models "appear" and somehow "perdure" in a decentralized evolution ?

Restrain the evolution to stick to the chosen model

Are those models somehow “tolerant” to dynamicity ?

Restrain the evolution to stick to the chosen model

Are those models somehow “tolerant” to dynamicity ?



Summing up studied models

Distributions	Cost to insert		Cost to leave	
	with size	w/o size	with size	w/o size
Erdős–Rényi Graphs	$\Theta(n)$	impossible	0	
Pairing multigraph (degree m)	$m/2 + \mathcal{O}(1/n)$		0	
Preferential attachment (degree m)	$m + 1 + \mathcal{O}(1/n)$		not distributed	
Uniform k-out graphs	$k + \mathcal{O}(1/n)$	$k\frac{e^2+3}{2} - 1 + \mathcal{O}(1/n)$	$0 + \mathcal{O}(1/n)$	$k + \mathcal{O}(1/n)$
Uniform μ-out graphs	$\lvert\mu\rvert + \mathbb{E}(\mu)$	$\mathcal{O}(\lvert\mu\rvert)$	$\mu(0) + \mathcal{O}(1/n)$	$\mathbb{E}(\mu) + \mathcal{O}(1/n)$

Conclusion

Models analysed

- Erdős–Rényi: unmaintainable without knowledge of n when p is fixed
- Pairing models: efficient maintenance without size needed
- Uniform k-out Graphs: interesting model with efficient maintenance without size needed

Conclusion

Models analysed

- Erdős–Rényi: unmaintainable without knowledge of n when p is fixed
- Pairing models: efficient maintenance without size needed
- Uniform k-out Graphs: interesting model with efficient maintenance without size needed

Many interesting open questions to investigate

- Maintability of Erdős–Rényi Random Graphs of varying density
- Full decentralized maintenance of Barabási–Albert model
- Other graph distributions (geometrical graphs, etc)
- What about approximate maintenance ?

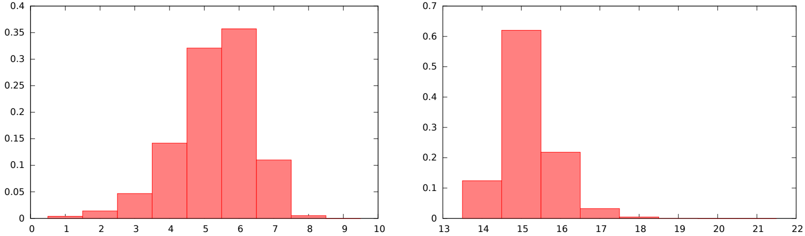
Thank you

Thank you for your attention.

Plan

1. **Uniform** k**-out Random Graphs**
2. Pairing and Barabási–Albert models

Why k-out random graphs ?



(a) Distances for 10^3 nodes (b) Max degree for 10^7 nodes

Figure : More statistics for k-out random graphs.

Plan

1. Uniform k-out Random Graphs
2. **Pairing and Barabási–Albert models**

Maintenance of the Barabási–Albert model

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

Maintenance of the Barabási–Albert model

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

Simulation of preferential attachment using `RandomVertex()`

- Sample a uniform vertex v using RV, then:
  1. keep v with probability $1/2$
  2. select uniformly one of its **outgoing** neighbours with proba $1/2$

Maintenance of the Barabási–Albert model

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

Simulation of preferential attachment using `RandomVertex()`

- Sample a uniform vertex v using RV, then:
  1. keep v with probability $1/2$
  2. select uniformly one of its **outgoing** neighbours with proba $1/2$

Correction and Cost

- Easy probability tricks :

$$\frac{1}{n}\frac{1}{2} + \sum_{u \in N^-(v)} \frac{1}{n}\frac{1}{2}\frac{\delta^{\to v}(u)}{k} = \frac{1}{2n} + \frac{\delta^-(v)}{2kn} = \frac{\delta(v)}{2kn}$$

- needs on average $k + \mathcal{O}(1/n)$ calls to RV

Maintenance of the Barabási–Albert model

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

Simulation of preferential attachment using `RandomVertex()`

- Sample a uniform vertex v using RV, then:
  1. keep v with probability $1/2$
  2. select uniformly one of its **outgoing** neighbours with proba $1/2$

Correction and Cost

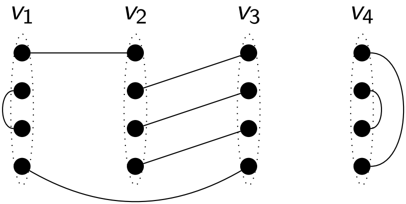
- Easy probability tricks :

$$\frac{1}{n}\frac{1}{2} + \sum_{u \in N^-(v)} \frac{1}{n}\frac{1}{2}\frac{\delta^{\to v}(u)}{k} = \frac{1}{2n} + \frac{\delta^-(v)}{2kn} = \frac{\delta(v)}{2kn}$$

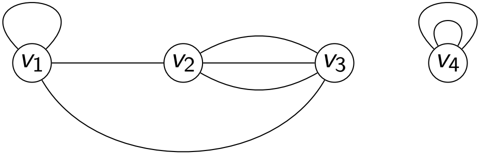
- needs on average $k + \mathcal{O}(1/n)$ calls to RV

To account of edges not yet present (while inserting a new vertex), we use a simple rejection mechanism.

Maintenance of the pairing model

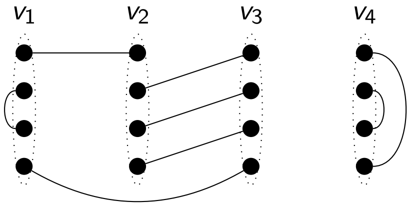


(a) Perfect matching M

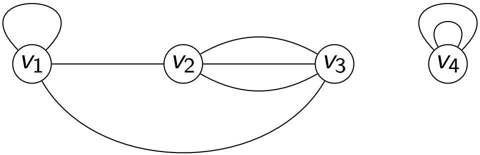


(b) Multigraph $P(M)$ of degree $k = 4$

Maintenance of the pairing model

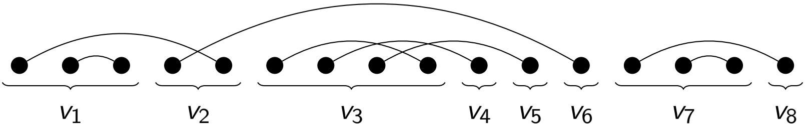


(c) Perfect matching M

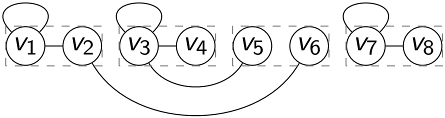


(d) Multigraph $P(M)$ of degree $k = 4$

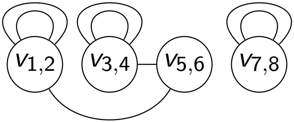
Only insertion needs $k/2 + \mathcal{O}(1/n)$ calls to RV, on average.



(e) Chord diagram with 16 points



(f) Constructed pseudograph



(g) after node merging

Figure : Example of Bollobás–Riordan construction.

Barabási–Albert model

Well known *scale-free* distribution used to model the Web, social networks, etc. Scale-free: proportion of node of degree k (for great k) is about $k^{-\gamma}$.

Barabási–Albert model

Well known *scale-free* distribution used to model the Web, social networks, etc. Scale-free: proportion of node of degree k (for great k) is about $k^{-\gamma}$.

Description

Upon arrival a node select a constant number of edges towards existing nodes using *preferential attachment* (with probability proportional to their degree in the graph).

Barabási–Albert model

Well known *scale-free* distribution used to model the Web, social networks, etc. Scale-free: proportion of node of degree k (for great k) is about $k^{-\gamma}$.

Description

Upon arrival a node select a constant number of edges towards existing nodes using *preferential attachment* (with probability proportional to their degree in the graph).

Explicit model – Bollobás–Riordan Multigraph model

$$\mathbb{P}(v_i = v) \quad = \quad \frac{\delta_i(v)}{2(kn+i)-1} \qquad \text{pour } v \in V$$

where k is the number of chosen neighbours during insertion and n is the size of the network.

Barabási–Albert model

Well known *scale-free* distribution used to model the Web, social networks, etc. Scale-free: proportion of node of degree k (for great k) is about $k^{-\gamma}$.

Description

Upon arrival a node select a constant number of edges towards existing nodes using *preferential attachment* (with probability proportional to their degree in the graph).

Explicit model – Bollobás–Riordan Multigraph model

$$\mathbb{P}(v_i = v) \quad = \quad \frac{\delta_i(v)}{2(kn+i)-1} \qquad \text{pour } v \in V$$

where k is the number of chosen neighbours during insertion and n is the size of the network.

To forget the insertion order dependencies, we consider it to me uniform: exchange of neighbourhoods after each insertion.

Maintenance of the Barabási–Albert model

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

Simulation of preferential attachment using `RandomVertex()`

- Sample a uniform vertex v using RV, then:
  1. keep v with probability $1/2$
  2. select uniformly one of its **outgoing** neighbours with proba $1/2$

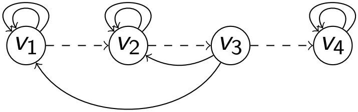
Maintenance of the Barabási–Albert model

Need to remember the orientation of the edges. This information can be recomputed but may need linear time to do so.

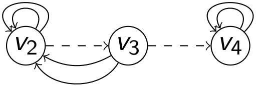
Simulation of preferential attachment using `RandomVertex()`

- Sample a uniform vertex v using RV, then:
  1. keep v with probability $1/2$
  2. select uniformly one of its **outgoing** neighbours with proba $1/2$

Deletion: needs to maintain a chain of the nodes and keep track of the "last inserted vertex".



(e) Augmented Multigraph



(f) After v_1's deletion

Barabási–Albert model conclusion

Maintenance of the model

- Efficient insertion is possible using extra structural information, but distributed deletion algorithms are not known
- Efficient maintenance is still open without extra information (edges orientation)
- Knowledge of n does not seem to help much

Barabási–Albert model conclusion

Maintenance of the model

- Efficient insertion is possible using extra structural information, but distributed deletion algorithms are not known
- Efficient maintenance is still open without extra information (edges orientation)
- Knowledge of n does not seem to help much

Alternative model: Pairing model

- based on uniform pairing (as Barabási–Albert, but the construction of the final graph is different)
- is used to prove results on scale-free graphs