Networks Structure and Dynamics
9. Dynamical properties of networks

Propriétés dynamiques

Maximilien Danisch, Fabien Tarissan

LIP6 – CNRS and Université Pierre et Marie Curie

first_name.last_name@lip6.fr

Decembre 5^{th} 2017

Plan

Outline

1. Introduction
2. Paths, reachability and temporal distances
3. Observations – Rollernet
4. Reachability graphs
5. Dynamic centralities

Motivations

Almost all networks are dynamic. Which dynamic?

Motivations

Almost all networks are dynamic. Which dynamic?

Dynamic OF the network

Appearing and disappearing of

- nodes
- links

Dynamic ON the network

- diffusion of viruses
- sending messages

Question

How to describe properly the dynamics?

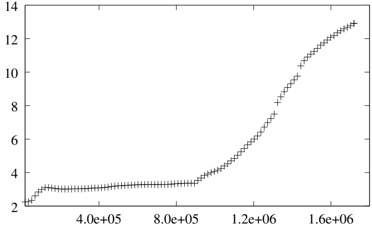
Classical approach

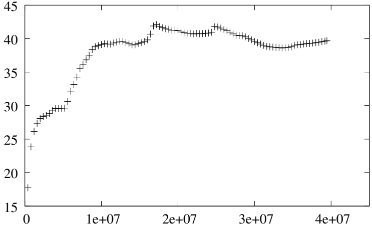
Rely on previous metrics used to describe static networks

- degree
- clustering
- communities
- ...

Example

Evolution of the average degree for two different graphs





$\longrightarrow$ Provide meaningful information

... but it is not enough. Why ?

Inconvenients

Lack of properties dealing truely with dynamics

- How long last the nodes/links
- Temporal paths
- ...

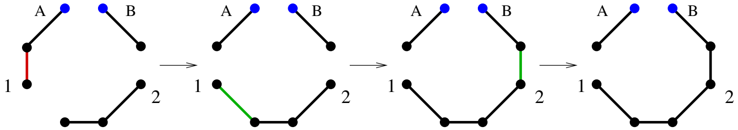
Different types of dynamics

Two major approaches providing temporal insights

- Periodic measures (eg. a radar)
- Record of temporal events (eg. exchanges of emails)

Plan

Temporal paths



Path from A to B
No path from B to A

Strong difference with the static version of distance

Temporal path – definition

Définition intuitive

Succession of nodes $u_1, \ldots, u_k$ such that:

- there exists a link (u_1, u_2) at time t_1, (u_2, u_3) at time t_2, ...
- $t_1 < t_2 < \ldots$

Variants

Several variants according to different authors:

- The use of a link is immediat
(nb of links one can use at a given time is then infinite)
- One need δ to go through a link
(given as a parameter)

More or less reallistic according to the context
More or less easy to compute

Several definitions rely on paths, such as ...?

Distance

Distance

At least 3 natural definitions:

- the least number of hops
- shortest in time to reach the target node
- Fastest when transfer begins

All the notions are useful depending on the context

We will focus on the shortest in time to reach the target node

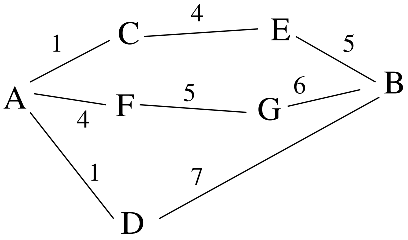
Distance – definition

Définition

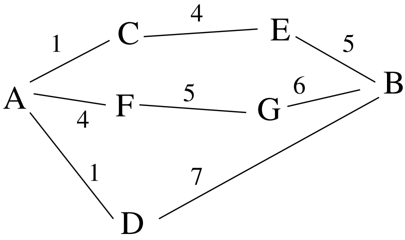
Let i, j be nodes, t a starting time.
Let t_a be the smallest time one needs to send a message from i to j (if possible)

Then the *temporal* distance from i to j at time t is: $t_a - t$

Distance – examples



Distance – examples

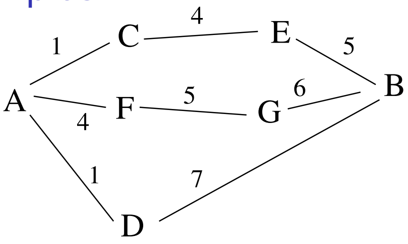


Distance de A à B

- Shortest in time to reach B: A – C – E – B
- Less number of hops: A – D – B
- Fastest when transfer begins: A – F – G – B

The distance depends on the starting time!

Distance – examples

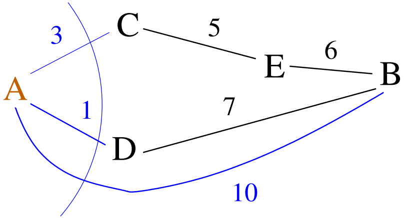


Distance de A à B

- Shortest in time to reach B: A – C – E – B
- Less number of hops: A – D – B
- Fastest when transfer begins: A – F – G – B

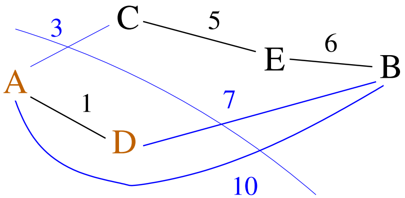
The distance depends on the starting time!
Exercise: ideas to obtain shortest (in time) from *s* to all nodes.

Distance – computation



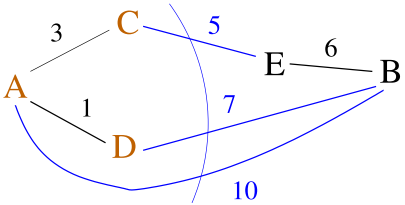
A 1

Distance – computation



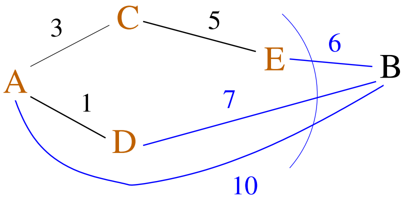
A 1
D 1

Distance – computation



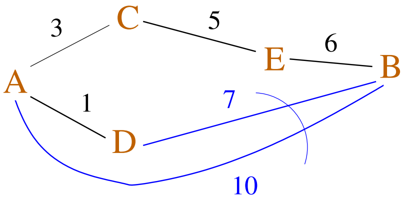
A 1
D 1
C 3

Distance – computation



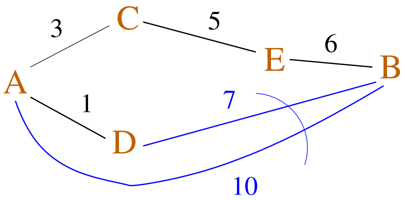
A 1
D 1
C 3
E 5

Distance – computation



A 1
D 1
C 3
E 5
B 6

Distance – computation



A 1
D 1
C 3
E 5
B 6

Exercise: Formalize the algorithm to obtain shortest (in time) from *s* to all nodes.

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s, x, t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$
While $Q \neq \emptyset$:
    take the closest link to the border s.t. $t_{cur} \leq t$ (u, v, t)
    $t_{cur} = t$
    If v unmarked:
        mark v as reached at time t
        $\forall (v, x, t) \in E$, add (v, x, t) to Q

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s,x,t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$
While $Q \neq \emptyset$:
  take the closest link to the border s.t. $t_{cur} \leq t$ (u,v,t)
  $t_{cur} = t$
  If v unmarked:
    mark v as reached at time t
    $\forall (v,x,t) \in E$, add (v,x,t) to Q

Complexity

While : look at (almost) all links: m

maintaining Q : $\sim \log m$

$\sim m \log m$ steps

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s, x, t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$
While $Q \neq \emptyset$:
  take the closest link to the border s.t. $t_{cur} \leq t$ (u, v, t)
  $t_{cur} = t$
  If v unmarked:
    mark v as reached at time t
    $\forall (v, x, t) \in E$, add (v, x, t) to Q

Complexity

While : look at (almost) all links: m

maintaining Q : $\sim \log m$

$\sim m \log m$ steps
... and for any node to any node?

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s, x, t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$
While $Q \neq \emptyset$:
  take the closest link to the border s.t. $t_{cur} \leq t$ (u, v, t)
  $t_{cur} = t$
  If v unmarked:
    mark v as reached at time t
    $\forall (v, x, t) \in E$, add (v, x, t) to Q

Complexity

While : look at (almost) all links: m

maintaining Q : $\sim \log m$

$\sim m \log m$ steps

... and for any node to any node? ... and for any node to any node starting at any time?

Better algorithm

Computation in two steps:

- For all pairs of nodes
- For all starting times

One memorizes the arrival time of a message
(instead of the distance)

Idea

- Suppose known all the arrival times for all starting times $> t$
- A link (u, v) at time $t - 1$ leads to the following:
  - u and v can be reached with distance 0 at time $t - 1$
  - For all node $x \neq u, v$
    - d_{ux} : temp. dist. from u to x, d_{vx} : temp. dist. from v to x
    - si $d_{ux} < d_{vx}$ then v can use u to reach x sooner
    - and vice-versa.

Algorithm

Sort the link by decreasing order of time

One uses two matrices $n \times n$: `dist` and `prev_dist`
(initialized at ∞: impossible to send a message)

`t_cur` = current time

`dist[x][x] = t_cur, prev_dist[x][x] = t_cur` for all `x`

Algorithme (2)

```
For all link (u,v) at time t
    If t != t_cur
        Copy dist in prev_dist
        cur_t = t
        dist[x][x] = t_cur for all x
    dist[u][v] = cur_t, dist[v][u] = cur_t
    For all x != u,v
        If prev_dist[u][x] != ∞ and prev_dist[v][x] != ∞
            If dist[u][x] > prev_dist[v][x]
             dist[u][x] = prev_dist[v][x]
            Else, if dist[v][x] > prev_dist[u][x]
             dist[v][x] = prev_dist[u][x]
        Else, if prev_dist[u][x] != ∞ and dist[v][x] > prev_dist[u][x]
            dist[v][x] = prev_dist[u][x]
        Else, if prev_dist[v][x] != ∞ and dist[u][x] > prev_dist[v][x]
            dist[u][x] = prev_dist[v][x]
```

Plan

Contact networks *Rollernet*

Tournoux et al. - *INFOCOM, 2009*

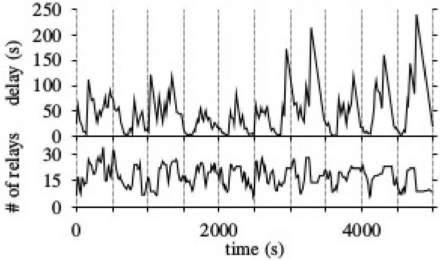
Measurements

- Measure the communication capabilities among individials
- Each participant is equipped with a bluetooth device
- Periodic record of the neighborhood

Rollerblade tour experiment:

- 62 nodes
- 180 minutes

Average delay



Modification of the delay, due to the regular stops (and the beggining of the tour)
Impact on the quality of the communications

Plan

Temporal reachability graphs

Whitbeck et al. - *MOBICOM, 2012*

Goal: better understanding of dynamic graphs

Reachability graph: directed dynamical graph

Definition: $R_\delta(t)$

Given δ :

$(u, v) \in R_\delta(t)$ if there exists a path from u to v :

- starting at time t
- arriving before $t + \delta$

Note: assumption of the article:

following a link takes time (τ).

Dataset

Rollernet

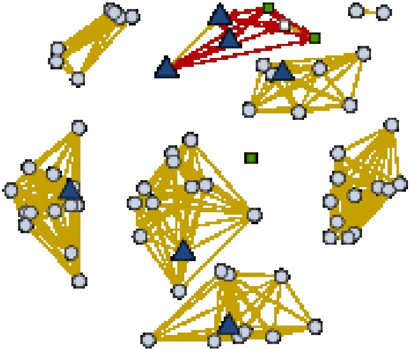
See previous slides

Stanford

- One day in high school
- 782 nodes

Observations

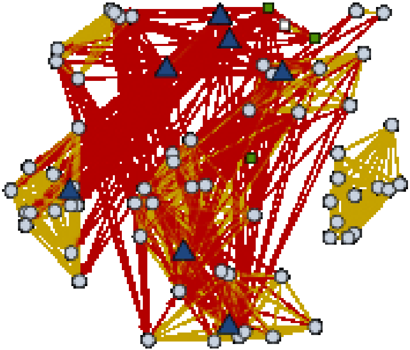
Stanford dataset – $\delta = 20$ minutes



Triangles: professors
Circles: students
Dark red arrows: asymmetric arcs

Observations

Stanford dataset – $\delta = 40$ minutes



Triangles: professors
Circles: students
Dark red arrows: asymmetric arcs

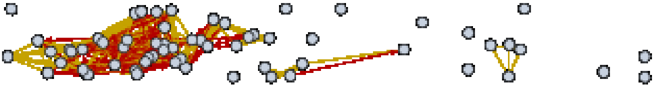
Comments

Observation

Study the temporal reachability graph allows to detect coherent groups
(with a well chosen δ)

Observations

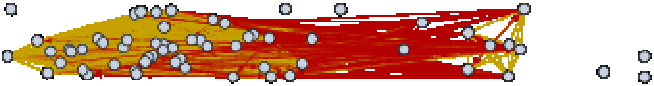
Rollernet dataset – $\delta = 10$ secondes
Acceleration phase



Red links: asymmetric links

Observations

Rollernet dataset – $\delta = 60$ secondes
Acceleration phase



Red links: asymmetric links

Comments

Observation

Impossible to send a message from the tail to the head.
Slow communication from head *towards* the tail is possible
Strong asymmetry

Plan

Centrality measures

Question: how to quatify the importance of a node in a network?

Centrality measures emphisize the role of a node as key node to relay information.

Different metrics highlights differents properties:

- Degree centrality
- Katz centrality
- Betweeneness centrality
- Closeness centrality
- Eigevector centrality
- ...

Betweeness centralities

Let G be a graph and v, s and t be nodes of G.
We call *betweeness centralities* the value:

$$BC(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

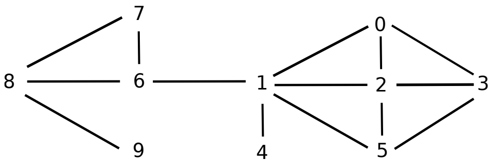
with:

- σ_{st} : nb of shortest paths between s and t
- $\sigma_{st}(v)$: nb of shortest paths between s and t going through v

Meaning

Captures the importance of a node v in a graph as a relay for diffusing information:

- propagation of signals/messages/virus
- connectivity of the network



Exercise : compute $BC(0)$, $BC(1)$ and $BC(4)$.

Algorithm

One needs to know **all** shortest paths between s and t.

Algorithm

One needs to know **all** shortest paths between *s* and *t*.

- BFS: 1 shortest path
- Needs to modify the algorithm. Idea:

Exercise : Compute the DAG of all shortest paths starting from 3 in the previous graph.

Algorithm

One needs to know **all** shortest paths between *s* and *t*.

- BFS: 1 shortest path
- Needs to modify the algorithm. Idea:
  - Compute a spanning DAG (Directed Acyclic Graph)
  - Rely on the computation of the distances
  - Use the depth of a node (distance to the root) in order to decide if a node is already part of the DAG or not.

Exercise : Compute the DAG of all shortest paths starting from 3 in the previous graph.

Algorithm

Exercice :

- Propose an algorithm computing the DAG of all shortest paths, given a graph G and a node s.
- Given a DAG of shortest paths, propose a formula allowing to compute, for every node v, the coouple (dn, up) où:
  - dn : nb of downward paths of v
  - up : nb of upward paths of v
- Deduce the number of shortest paths starting from s and going through v
- Apply the algorithm to the DAG starting from 3 in the previous graph
- Propose the final algorithm enabling to compute the betweeness centrality of a given node.

Betweeness centrailty

Which dynamic version for centrality?

Dynamic centrality – several propositions

Evolution of the standard centrality

Compute the betweeness centrality for each time intervals

- depends on the size of the window
- do not take into account reallistic communications in most of the cases

Extension to temporal paths

For a node *i*, compute the fraction of shortest temporal paths going through *i*.

- Temporal paths depends strongly on starting time

Temporal Betweenness Centrality

Nicosia et al. - *in* Temporal Networks*, 2013*

Take into account the waiting time on the nodes:
If one has a unique shortest time from i to j :
$i \longrightarrow k \longrightarrow j$

the importance of k depends on the time the message "spends" on k

Temporal Betweenness Centrality – definition

Nicosia et al. - *in* Temporal Networks, *2013*

Take into account the waiting time on the nodes:

$$C_i(t_m) = \frac{1}{(n-1)(n-2)} \sum_{j \neq i} \sum_{k \neq i,j} \frac{U(i, t_m, j, k)}{\sigma_{jk}}$$

- $U(i, t_m, j, k)$: nb of shortests temporal paths from j to k such that one uses i at a time $\leq t_m$
- σ_{jk} : nb of shortest temporal paths from j to k

Average centrality: average over all time instants

Temporal Betweenness Centrality – drawbacks

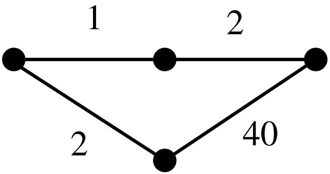
- All paths start at the initila time!
- Only the average value is studied

Mediation

Tang et al. - *in* Temporal Networks, *2013*

Idée

If k is on a shortest temporal path between i and j
its importance depends on the second shortest temporal path.



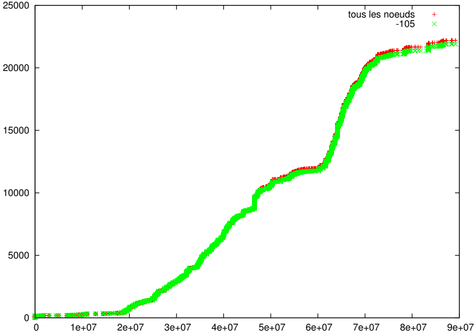
Mediation – in practice

Principe

- Compute all distances between all pairs of nodes
- Suppress the node *i*
- Compute again all distances
  Difference : importance of node *i*

Mediation - example

Cumulative distribution of distances



Need to take into account paths that start at all instants

Importance of a node – other propositions

Alternatives exist

- Closeness centralities and extensions

  Time Evolution of the Importance of Nodes in dynamic Networks, Magnien & Tarissan - *in* Asonam*, 2015*

- . . .

No consensus

Conclusion

Properties defined for staric networks are unsufficient to describe dynamics networks
In this course:

- Properties related to temporal paths

Other preperties

- Temporal patterns
- Duration of nodes/links
- Resilience of links
- Dynamic communities
- ...