Approaches for Quantifying Multicast Complexity

Nora Berg

AW 2

15. 5. 2014

Introduction

Motivation

- High complexity leads to low predictability
- Complexity varies in different algorithms

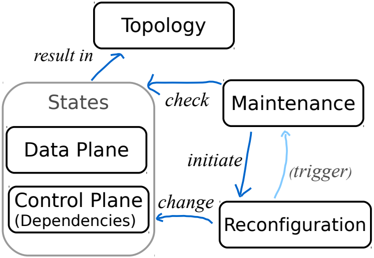
Long-Term Goal

- Identify (some) origins of complexity
- Analyze impact due to different multicast algorithm
- Describe (some) aspects of complexity in a formal way
- Predict weak resilience due to complexity

Control Loop

Basic Assumptions:

- Autonomous nodes
- Identical control loops



- States: knowledge of a node
- Maintenance: local operations to monitor consistency
- Reconfiguration: network operations changing the states (and the tree)

Aspects of Complexity

Robustness

How many messages are *delivered despite errors*?

Resilience

Do *repair* mechanisms exist?

How many options are there to repair the network?

Sustainability

How much effort is used to *keep* the network in a stable state?

Approaches

- Networks = Graphs $G(V, E)$
- *Problem:* Graphs are static and passive
- High complexity lead to unpredictable changes of the graph
- Introduced by protocols and algorithms
- Change is a function with two inputs:
  - Given underlying network (graph)
  - Current overlay network (multicast distribution tree)
- And one output:
  - New multicast distribution tree

$\Rightarrow$ 3 approaches to describe the change

Graph Transformation in a Nutshell (2010)

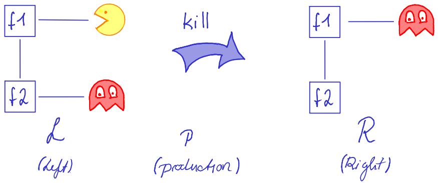
Title:
Graph Transformation in a Nutshell [5]

Author:
Reiko Heckel

Other Sources:
H. Ehrig, K. Ehrig, U. Prange, G.Taentzer [3]
Fundamentals of Algebraic Graph Transformation (2006)

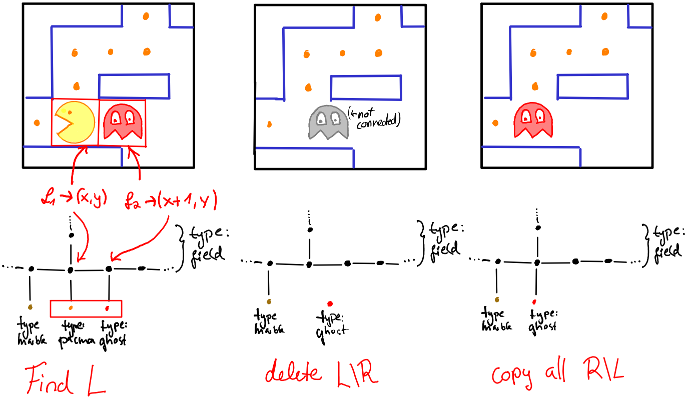
Basics

- Describes the *change of a graph* $G(V, E, s, t)$
- Possible changes are described as *rules*
- Sequence of production rules describes behavior of the model (e.g. a pacman game)
- 2 methods to describe rules
  - Single Pushout $p = (L, R)$
  - Double Pushout $p = (L, K, R)$



L: Precondition R: Postcondition

Single Pushout (SPO)



Algorithm for SPO-Graphtransformation $G \overset{p}{\Rightarrow} H$

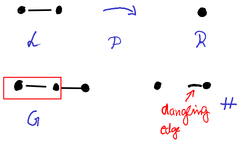
- *Find* an occurence of L in G
- *Delete* from G all vertices and edges matched by $L \setminus R$
- *copy* all $R \setminus L$ to H

$\rightarrow$ Typing needed in many applications (not only in SPO)

Problems of SPO

Dangling Edges

- SPO-Algorithm does not guarantee a valid graph
- Dangling edges possible

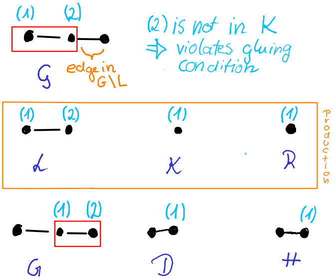


Solutions

- Delete all dangling edges from H
- Use gluing conditions with the *double-pushout* approach

Double Pushout (DPO)

- Production $p = (L, K, R)$
- Morphism m is injective structure-preserving mapping $L \to G$
  - adjacent vertices in G stay adjacent in H



Gluing Condition

Each vertex x in L, such that m(x) is source or target of an edge e in $G \backslash L$ must be in K.

Algorithm for DPO-Graphtransformation $G \Rightarrow H$

- *Find* an occurence of L in G which matches the *gluing condition* (create m)
- Create context Graph $D := (G \backslash m(L)) \cup m(K)$
- Create $H = R +_K D$

Short Reflection

What is graph transformation used for?

- Modelling of system states or structures
- Set of rules generate graphs: *graph grammar*
- Analyzeable with methodes of formal languages

How is graph transformation useable for complexity?

- Modelling reconfiguration methods as rules and states as types
- Is it possible to predict the occurence probability of a well known failure state with formal language methods?

Stability of a Multicast Tree

- Title: Stability of a Multicast Tree [7]
- Author: P. van Mieghem and M. Janic
- Other sources:
  P. van Mieghem: Performance Analysis of Communications Networks and Systems [6]
  P. Hartmann: Mathematik fr Informatiker [4]

Basics

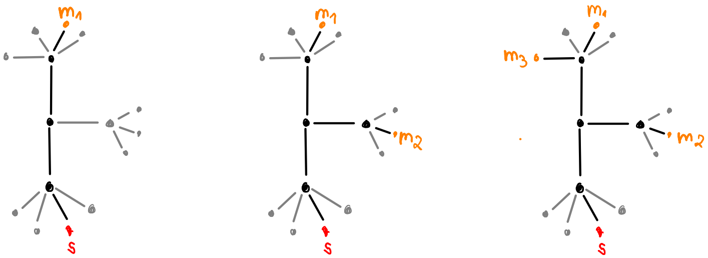
Question

How much does a multicast tree change, if a node joins?

Assumptions:

- Network Graph (RGU) $G_p(N)$
  - Random graph
  - Independently chosen links with propability p
  - Uniformly/exponent. distributed link weights ω
- Multicast tree:
  - Shortest path tree
  - *One* source
  - m receivers
  - Over network $G_p(N)$
  - N nodes in the network

Example: Changing Tree



[grey: Network, black: MC-Distribution Tree]

- joining nodes connect to the tree through shortest path
- with increasing amount of receivers the amount of new edges decreases

Approach

Stability

If a member leaves, on average *less than one link changes*

- $g_N(m)$:
  - Average number of *hops* in a *shortest path tree* with m receivers
- $\Delta_N(m)$:
  - Changed links on a multicast tree if a member leaves
- $E[\Delta_N(m)] = g_N(m) - g_N(m-1)$
  - expected (mean) change of a multicast tree

Results

Stability

- Stable, if multicast network has $\geq \frac{N}{3}$ receiver
- Problem: Unrealistic setting

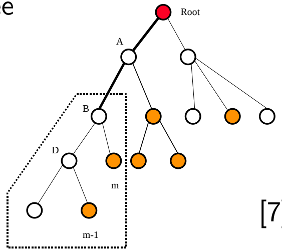
Change of a Network

- Sparse fill level
- Change of the multicast tree is a Poisson distribution

Change as Poisson distribution

Assumptions

- Network graph is RGU with N nodes
- Multicast tree is a uniform recursive tree
- Multicast tree is a shortest path tree
- Sparse fill level ($< \frac{N}{3}$)



[7]

Poisson Distribution

- Number of events
- Fixed interval (time, space,...)
- Independent from prior interval

General Form:

$$Pr(X = k) = \frac{\gamma^k}{k!} \cdot e^{-\gamma}$$

This Case:

$$Pr[\Delta_N(m) = k] \sim \frac{(E[\Delta_N(m)])^k}{k!} \cdot e^{-E[\Delta_N(m)]}$$

Simulation Results

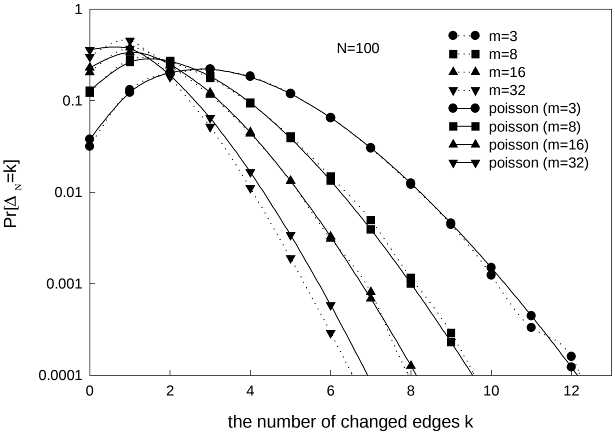


Fig. 5. SPT: Pdf $Pr[\Delta_N = k]$ for $N = 100$ and $m < N/3$

Source: P. van Mieghem, M.Janic; Stability of a Multicast Tree

Catastrophic cascade of failures in interdependent networks (2010)

Title:
Catastrophic cascade of failures in interdependent networks

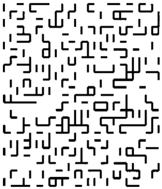
Authors:
S.V. Buldryref, R. Parshani, G. Paul, H.E. Stanley, S. Havlin

Other sources:
R. Albert, A. Barabasi:
Statistical mechanics of complex networks (2002) [1]
S. Havlin, R. Cohen:
Complex Networks. Structure, Robustness and Function (2010) [2]

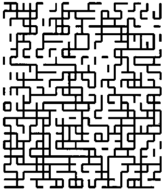
Percolation Theory

Motivation

- Interdependent networks
- Malfunctions lead to failures in coupled networks
- Example: Italian Blackout 2003



p=0.315



p=0.525

[image: [1]]

Percolation Theory

- Origins in physics & material science
- Grid structure of dimension d
- Links between neighbours are occupied with propability p
- Main Question: Is there a path from one side to the other?

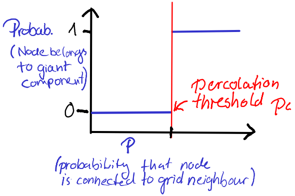
Percolation Theory on Networks

Percolation Theory

- Giant component: subgraph containing nearly all nodes
- Percolation threshold: p_c
- Small p: unconnected graph, many clusters
  - probability of a node being in the giant component $\rightarrow 0$
- $p \geqslant p_c$ One giant component emerges
  - probability of a node being in the giant component $\rightarrow 1$
- Exponential decay of the cluster size

On Networks

- Difference in degree distributions:
  - Grid: max. 2d-Neighbours
  - Network: higher node degrees
- Solution: ∞-dimensional grid



Approach for Interdependent Networks

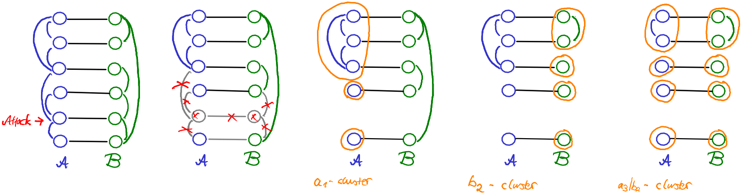
Given:

- Two Networks A and B of size N
- Each Node A_i is coupled to equivalent node in B_i
- Within A and B: degree distribution $P_A(k) = P_B(k)$

Bidirectional Link: $A_i \leftrightarrow B_i$

Node A_i depends on functionality of Node B_i

Subsequent Removal of Links



Idea

- Remove a fraction of $(1 - p)$ nodes

→ Remove the bidirectional coupled nodes

→ Remove stepwise links within the network B, which connects different cluster of Network A (or vice versa)

Main Question in the Paper:

- Analyze the size of p, when the network breaks into pieces
  - → Percolation threshold: p_c
- Influence of the node degree distribution

Results

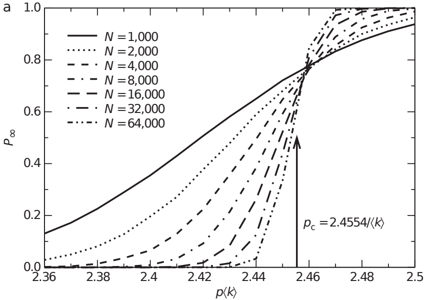
Dependency of p_c on Degree Distribution

- Single networks:
  p_c decreases with broader degree distribution
  → More stable with broader degree distribution

- Interdependent networks:
  p_c increases with broader degree distribution
  → Less stable with broader degree distribution
  → Probability increases to affect hub [1]
  → Hubs become vulnerability

[1] hubs: nodes with exceptional large node degree

Simulation Results



Further Results

- Analytical calculation of p_c
- Dependency: random graph type $\leftrightarrow$ p_c

Summary

- Overview on approaches which describe changes of graphs
- Graph Transformation can be used to model multicast algorithms
- The amount of new hops in multicast trees follows a Poisson distribution in networks with few members
- Many realistic networks are interdepent
- Stability of interdependent networks differs from single networks

So, what's next?

Possible next steps

- Extracting complexity critera from reconfiguration methods
- Testing them on simple example graphs (line, full mesh, etc.)
- How general is the SPT picture of distribution trees?
- What are the exceptions? And why?
- Decay of MC-networks *vs.* repair mechanisms
    - Which critera benefit from decay/repair mechanisms?
    - Which one is faster?

Thank you for your Attention!
Any Questions?

Bibliography I

[1] Réka Albert and Albert-László Barabási.
Statistical Mechanics of Complex Networks.
Reviews of modern physics, 74(1):47, 2002.

[2] Reuven Cohen and Shlomo Havlin.
Complex Networks: Structure, Robustness and Function.
Cambridge University Press, 2010.

[3] Hartmut Ehrig, Karsten Ehrig, Ulrike Prange, and Gabriele Taentzer.
Fundamentals of Algebraic Graph Transformation, volume 373.
Springer, 2006.

[4] Peter Hartmann.
Mathematik für Informatiker: Ein praxisbezogenes Lehrbuch. Vierte Auflage, 2006.

Bibliography II

[5] Reiko Heckel.
Graph Transformation in a Nutshell.
Electronic Notes in Theoretical Computer Science, 148:187–198, 2006.

[6] Piet Van Mieghem.
Performance Analysis of Communications Networks and Systems.
Cambridge University Press, Cambridge, New York, 2006.

[7] Piet Van Mieghem and Milena Janic.
Stability of a Multicast Tree.
In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1099–1108. IEEE, 2002.