# Evaluating network reliability versus topology by BDD algorithms

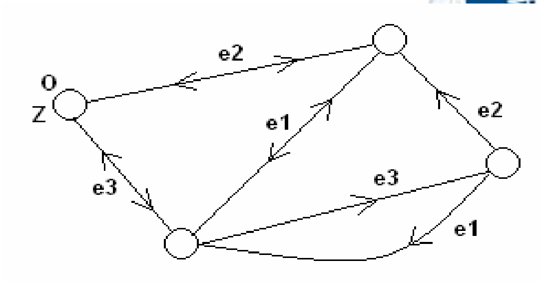
A. Bobbio, R. Terruggia, E. Ciancamerla, M. Minichino

PSAM 9 – International Probabilistic Safety Assessment and Management Conference
Hong Kong – China, 18 -23 May 2008

# Summary

- A network is a structure where any couple of nodes is normally connected by different independent paths, thus making the structure intrinsically dependable
- Network reliability emerges as a critical issue for characterizing a network, but it is also a major challenge (the problem is known to be NP-complete, thus no fast general algorithm is likely to exist)
- Binary Decision Diagrams (BDD) have provided an extraordinarily efficient technique to encode Boolean functions
- We compute network reliability relying on two BDD algorithms
- A preliminary tool is in progress
- Critical Infrastructures (CI) form a framework of interconnected and interdependent networks (well represented by two specific topological classes: Random Graphs and Scale Free networks)
- We present preliminary results of a scalable benchmark, that includes different network structures, namely Random and Scale Free networks

# Network definitions




- **Graph** G=(V,E): has two elements, nodes and arcs.
- **Binary network**: graph elements as binary objects (working/failed states)
- **Connectivity**: is defined by means of the **paths** connecting any two nodes – in a binary network connectivity becomes a boolean function
- **Reliability** may be computed, if a probability measure (of being up/down) is associated to elements of the network
- **Two-terminal reliability** is the probability that two nodes communicate with each other by at least one path of working edges: (s,t) – reliability

3

# Network reliability

Arcs and nodes are binary entities

We study:
- **Connectivity**
- **Reliability**

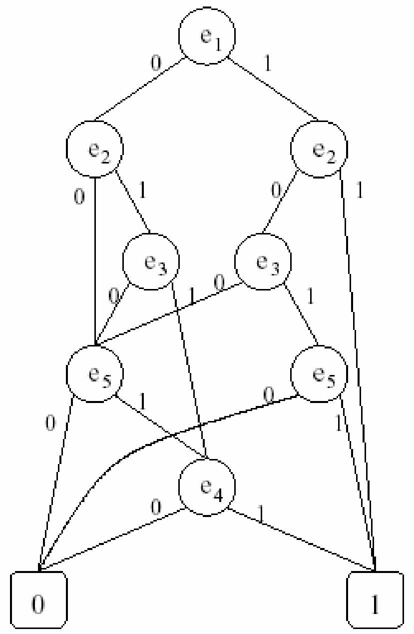
- Qualitative analysis:
    - **Minimal paths**
    - **Minimal cuts**

- Quantitative analysis:
    - **Reliability and Unreliability functions**

- **We show how the (s,t)-connectivity of a binary graph can be encoded into a BDD, and how the corresponding reliability measure can be computed for different network topologies**

# Binary Decision Diagrams

- A BDD represents a Boolean function by means of the Shannon's decomposition:
  - If $F$ is a Boolean function on the variables $x_1, x_2, \ldots, x_n$ the Shannon decomposition holds:
  - $F = x_1 \wedge F_{x1=1} \vee \underline{x}_1 \wedge F_{x1=0}$ - where, ($\underline{x}_1 = NOT\, x_1$);
  - $F_{x1=1}$ is derived from $F$ assuming $x_1$ is true, and $F_{x1=0}$ is derived from $F$ assuming $x_1$ is false.
  - Applying iteratively the Shannon's decomposition formula pivoting with respect to a sequence of all the variables, a complete decomposition can be obtained.
- The sequence of decompositions can be represented in graphical form using a binary tree.
  - Each node of the tree represents the pivot variable with respect to which the decomposition is done.
  - Each node has two branches: the right branch has the value 1 and the left branch has the value 0
  - The BDD has a single root (represented by the first pivot variable) and two terminal leaves: the 1 leaf and the 0 leaf.
  - Any path from the root to the 1 leaf of a BDD represents a combination of variables (either direct or negated) that makes the function true (viceversa for a path from the root to the 0 leaf)



## ROBDD

- In the construction of the BDD the variables must be Ordered
- Occasionally, the binary tree contains identical subtrees.
- Reduction – Identical portions of BDD are folded
- The result is the Reduced Ordered BDD

# Network reliability via BDD

If we assign to every variable $x_i$ a probability $p_i$ of being true (1 - $p_i$ false), we can compute the probability $P\{F\}$ of the function $F$ by applying recursively the equation:

$$P\{F\} = p_1 P\{F_{x1=1}\} + (1 - p_1) P\{F_{x1=0}\}$$

Two different algorithms have been implemented to compute network reliability via BDD:

- the first based on the search of the minpaths
- the second based on a recursive visit of the graph

# 2-terminal reliability via Minpath Analysis (MPA)

- Given a network G=(V,E) and two nodes (s, t)
  - a (s, t) **Path**: a subset of elements (arcs and/or nodes), that guarantees the source and the destination to be connected if all the components of this subset are functioning.
  - a path is minimal (**minpath**) if does not exist a subset of nodes that is also a path

- If $H_1, H_2, \ldots, H_n$ are the minpaths between s and t, then the two terminal network connectivity S :
  - $S = H_1 \vee H_2 \vee \ldots \vee H_n$
  - The two terminal reliability $R_{s,t}$ can be determinated from network *minpaths*:
    - $R_{s,t} = \Pr\{S\} = \Pr\{H_1 \vee H_2 \vee \ldots \vee H_n\}$

- The connectivity expression is a Boolean function for which the Shannon's decompostion can be applied and the related BDD constructed.
  - From the BDD, the (s,t)-reliability can be finally computed from *P {F}* formula

## 2-terminal reliability by graph Visiting Algorithms (VA)

- The BDD representation of the 2-terminal connectivity of a graph, can be directly derived without passing from a preliminary search for the minpaths
- In literature an algorithm is proposed that generates the BDD directly, via a recursive visit on the graph
  - Given a graph $G = (V;E)$ and two nodes (s,t), the algorithm starts from s node and visits the graph (according to a given but arbitrary visiting strategy) until t node is reached
  - The BDD construction starts recursively once the sink node t is reached
  - The BDD's of the nodes along a path from s to t are combined in AND, while if a node possesses more than one outgoing edge the BDDs of the paths starting from each edge are combined in OR
  - As a last step the algorithm visit the source node and builds the BDD for the complete connectivity function
- The minpaths provide a qualitative information about the most favorable connections (in term of number of hops) between source and sink

# Tool implementation

- A software tool for network reliability analysis is under construction.
- The tool accepts in input a graph with various formats: incidence matrix, adjacency list, formats provided by other tools
- The user may optionally choose which elements are failure prone and assign the corresponding failure probabilities.
- Finally, the s and t node must be assigned so that the program can provide the two-terminal reliability and the list of minpaths.
- The tool embodies the two approaches previously described:
  - the preliminary search of the minpaths. The algorithm relies on a recursive call of a function based on the classical Dijkstra's algorithm. Once the minimal paths between nodes s and t are explored (and ordered by rank), the BDD is constructed.
  - the BDD for the chosen connectivity function is directly constructed. The construction and manipulation of the BDD's is managed through the BDD library developed at the Carnagie Mellon University.
  - the list of the minpaths and of the mincuts may be optionally determined.

10

## Scalable Benchmark

- A scalable benchmark is underway to evaluate and to compare the efficiency of the software tool against the increasing complexity of different network topologies: namely RG and SF networks.

# Structural properties of a Graph and measures

- The main structural properties of a graph stem from its classification as belonging to a specific topological class (i.e. Random and Scale Free)

- Structural properties can be characterized by different measures. We consider:

  - **the distribution of the connectivity degree**: by defining as connectivity degree $k$ of a node the number of arcs emerging from that node, the distribution of connectivity degree $P(k)$ is the probability of a node to have degree $k$

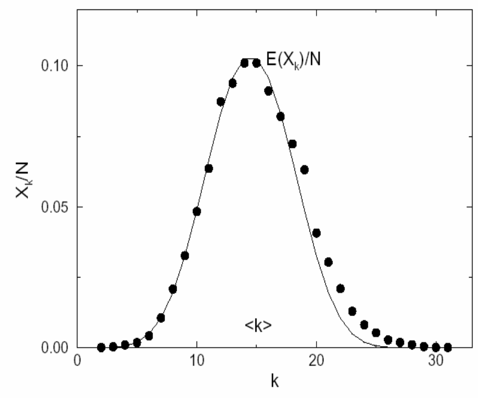
  - **the clustering coefficient** ***C***: measures the propensity of nodes to form local communities; the clustering coefficient is a measure of the fraction of neighbors of a node which are also neighbors each other.

- **From what concerns reliability studies on networks, the most relevant issue is that the structure of the network influences its reliability**

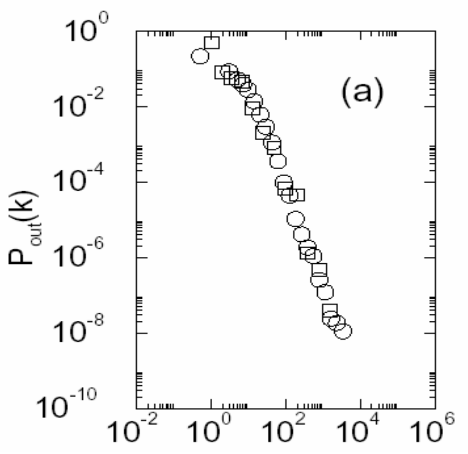
# Large Scale Networks

Structure and properties of complex networks

Random Graphs – The degree distribution is Poisson;




Scale free networks - The degree distribution is power law (long tail)




13

# Random and Scale Free networks

- The most relevant topological classes of networks for representing CI structures, are RG and SF networks.

  – *In RG networks*, the degree distribution $P(k)$ follows a Poisson distribution; the network degree is characterized by an average value $< k >$ with a given standard deviation. RG networks are believed to be produced by a growth mechanism where new nodes stuck randomly to existing nodes (random growth mechanism).

  – *In SF networks*, the degree distribution $P(k)$ follows a power-law. SF networks result in the simultaneous presence of a small number of very highly connected nodes (the hubs) linked to a large number of poorly connected nodes (the leaves). The growth model, known as *preferential attachment.* is realized by assuming that the probability $Pi(n + 1)$ of the $(n + 1)$-th generated node sticks to node $i$, increases linearly with the degree of $i$.

- SF networks are more robust to random arc or node removal than RG networks but are more prone to malicious attacks, where the nodes with highest degrees are first removed.

# Network generating algorithms

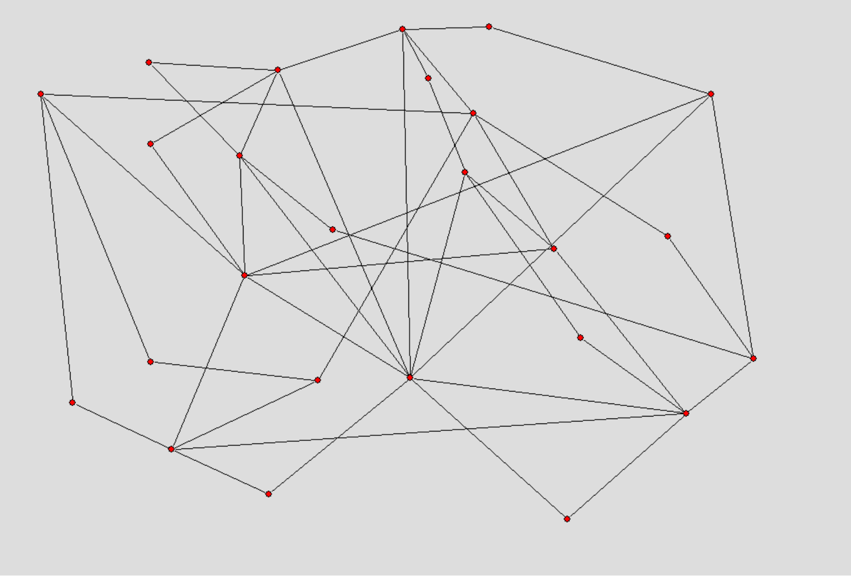
- to test the tool we have implemented two algorithms to respectively **grow a RG and a SF network**.

  - The growth mechanism of RG requires three input parameters: the number of connections (or degree $k$) that a newly generated node may establish; the probability of attachment $p$; and the final dimension of the network $N$.
  - Two random generators are used to establish the attachment of the $n$+1 node: a first one sorts a node $i$ at which the $n + 1$ node could be attached. A second generator sorts a random number $p_1$. If $p_1 <= p$, then$n$+1 node will be attached to node $i$
  - To generate a SF network, what changes is the attachment process. A first generator sorts a node $i$ at which the $n$+1 node could be attached. Node $i$ will be considered for the attachment if it has not been already sorted; in that case we compute $Pi(n+1)$ according to Equation 1. Then a second generator sorts a random number $p1$. If $p_1 <= Pi(n + 1)$ then the new $n + 1$ node is attached to node $i$.

- The growth algorithm is incremental, in the sense that once we have generated a graph with $N$ node of any topology, we can add on the same structure an arbitrary number of new nodes.

# Benchmark results on Random Graphs

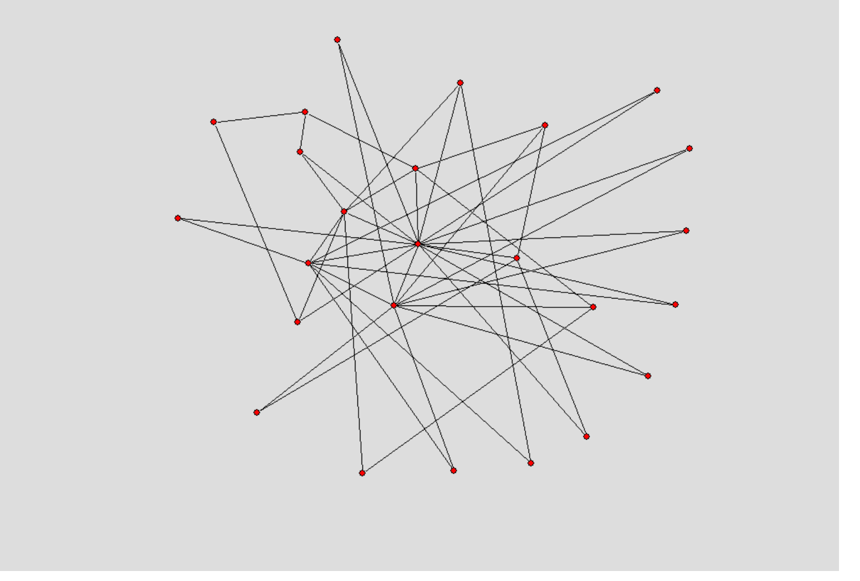
- Table displays the results obtained on RG networks
- Networks are grown according to the described algorithm with an increasing number of final nodes *N*, while keeping constant the number of connections ($k = 2$), and the probability of attachment ($p = 1$).
- The ¯first generated node is assumed as the source s and the last generated node as the sink t.
- The firrst three columns report the ¯final number of nodes, the final number of edges and the clustering coefficient. Columns 4, 5, 6 report, respectively, the number of minpaths, the number of the nodes of the BDD with VA algorithm, the number of the nodes of the BDD with MPA algorithm and the reliability value.



| #nodes | #arcs | clustering coefficient | # minpath | # BDD nodes VA | # BDD nodes MPA | (s,t) reliability |
|---|---|---|---|---|---|---|
| 20 | 72 | 0.1632 | 2046 | 316360 | 10032 | 0.98885 |
| 25 | 92 | 0.1267 | 21040 | 3333930 | 32653 | 0.98886 |
| 30 | 112 | 0.1502 | 119033 | *n.a.* | 708801 | 0.98906 |
| 40 | 152 | 0.1379 | 6757683 | *n.a.* | *n.a.* | *n.a.* |
| 50 | 192 | 0.0660 | *n.a.* | *n.a.* | *n.a.* | *n.a.* |

# Benchmark results on Scale Free

- Table displays the results obtained on SF networks
- Networks are grown according to the described algorithm with an increasing number of final nodes *N*, while keeping constant the number of connections (*k* = 2)
- The first generated node is assumed as the source s and the last generated node as the sink t.
- The firrst three columns of each table report the ¯final number of nodes, the final number of edges and the clustering coefficient. Columns 4, 5, 6 report, respectively, the number of minpaths, the number of the nodes of the BDD with VA algorithm, the number of the nodes of the BDD with MPA algorithm and the reliability value.



| #nodes | #arcs | clustering coefficient | # minpath | # BDD nodes VA | # BDD nodes MPA | (s,t) reliability |
|---|---|---|---|---|---|---|
| 20 | 74 | 0.5869 | 263 | 10409 | 266 | 0.98010 |
| 25 | 94 | 0.4388 | 2651 | 186350 | 5019 | 0.97815 |
| 30 | 114 | 0.3927 | 4707 | *n.a.* | 239788 | 0.98001 |
| 40 | 154 | 0.3888 | 73680 | *n.a.* | *n.a.* | *n.a.* |
| 50 | 194 | 0.3499 | *n.a.* | *n.a.* | *n.a.* | *n.a.* |

## Conclusions and future work

- A tool for the reliability analysis of networks by means of different independent algorithms via the construction of a BDD is under experimentation.
- The ability of the algorithms to cope with different network topologies of increasing complexity is under testing
- Network reliability problem is NP-complete
- For very large networks new algorithms are needed
- In the immediate future we are investigating:
  - the limits of the technique with extensive amount of static memory
  - the service availability of interconnected networks by convenient heterogeneous stochastic techniques,that include this one