Peer-to-Peer Networks

Practical Exercise 3

Dominik Fischer

P2P Networks Group at TU Darmstadt

27. November 2012

Exercise Evaluation

- Make messages reliable
  - acknowledge messages
  - sequence number & sender
  - or random identifier
  - be selective, only use when needed
- Manage the process
  - To start with one peer and
  - grow a network automatically, repeatedly
    - pick a random peer,
    - let it spawn or leave at random.

Exercise Evaluation

- How to spawn a peer?
  - create a thread
  - spawn a process
  - fork the program
- How to control the network?
  - central server
  - self-made broadcast
  - autonomous peers
  - native broadcast or multicast
- over time, networks get complex

netcat

- TCP frontend
- speaks UDP with parameter `-u`
- links command line input and output to network sockets
- may be called `nc` on some systems
- Server: `nc -ul <port>`
- Client: `nc -u <host> <port>`
- `man netcat`

Wireshark

- so called packet sniffer
- graphical tool to analyze network traffic
- Can decompose many protocols
- on a wide range of levels.
- reverse engineering tool

iftop / nethogs

- inspired by `top`, the cli task manager
- monitors network traffic by
  - program (iftop)
  - interface (nethogs)
- can determine sources and sinks
- identifies bottlenecks

graphviz

- graph layout tool, an easy way to display your network
- Simple input format
  - needs little more than edges in text format,
  - but bears possibilities for complex options.
- let all clients output their links and compose graph
- different layouts possible through a number of algorithms
- `man dot`

graphviz

graph.dot

```
digraph {
    a → b → c → f;
    c → d → f;
    b → d → e [color=red];
    a → g → h;
    x → w;
    node [shape=box];
    a → x → y → z;
}
```

graphviz

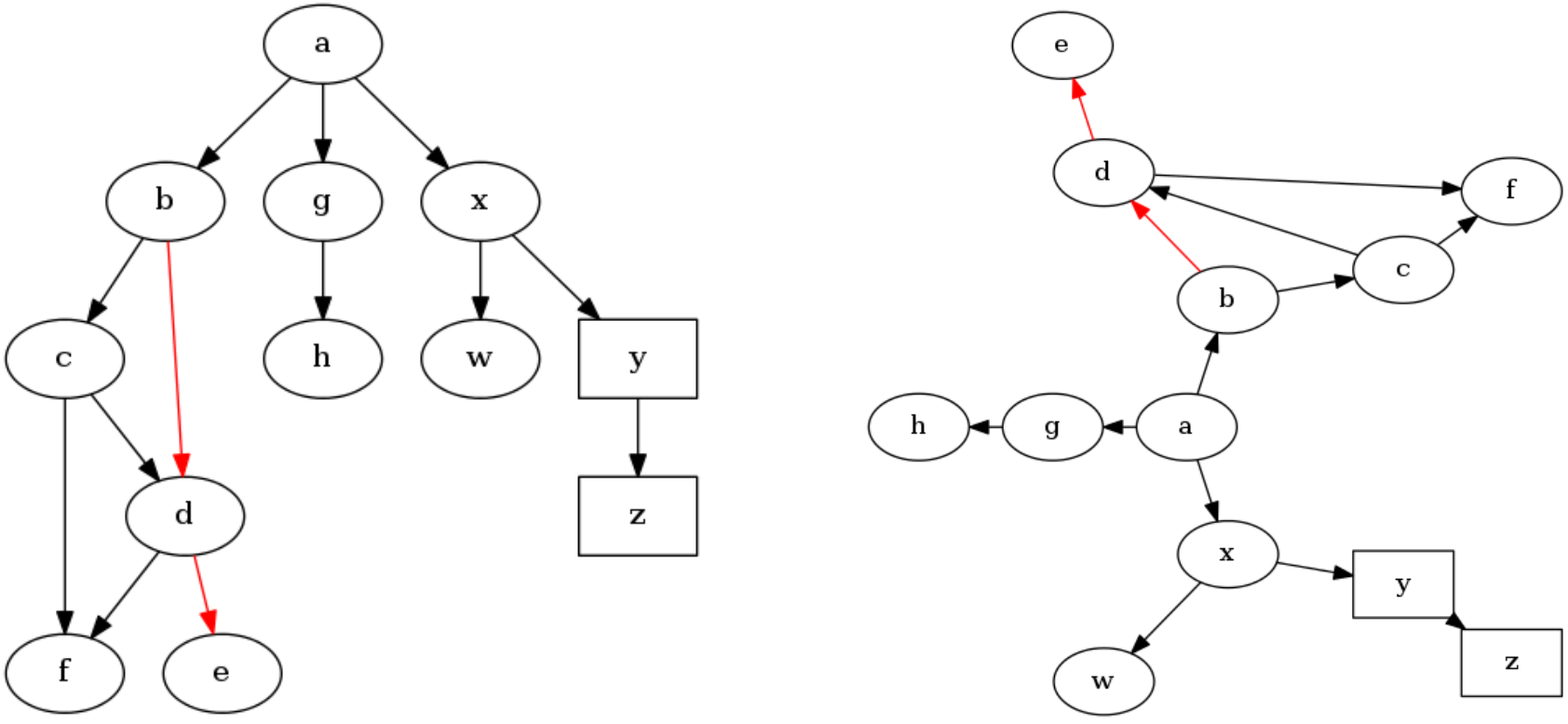

Figure : graph in dot and twopi layout

graphviz

- bindings for various programming languages
- Gephi provides a graphical frontend.

Graph visualisation

- various graph frameworks are available for all popular languages
- Java
  - Jung
  - JgraphT
- Python
  - NetworkX
  - igraph
- GraphViz binds with C directly.
- C++ has the Boost library.
- .NET
  - Graph#
  - QuickGraph

Programming Exercise 3

- Extend the Growth Peer with Broadcast.
  - Implement a message that reaches whole network
  - by flooding unlimited avoiding duplicates.
  - gain access to whole network structure
  - one peers can ask for all links of all nodes
  - think of iterative vs. recursive answers
  - central authorities may be replaced by broadcast
  - each node could control the network

Programming Exercise 3

- Analyze the Growth Peer network.
  - Experiment with parameters.
    - different chances of spwan vs. leave
    - different probabilities for peers to be chosen
    - compare leave handling (link all vs. circularly)
  - Visualize or analyze the network.
    - diameter, clustering, degree, ...
    - over time
    - by parameter configuration

Programming Exercise 3

- 20 points to achieve
  - 10 for broadcast
  - Bring a handful of findings to present at the attestation.
  - 2 – 5 points per analysis or visualisation
  - combinations possible
  - last task dealing with the growth peers