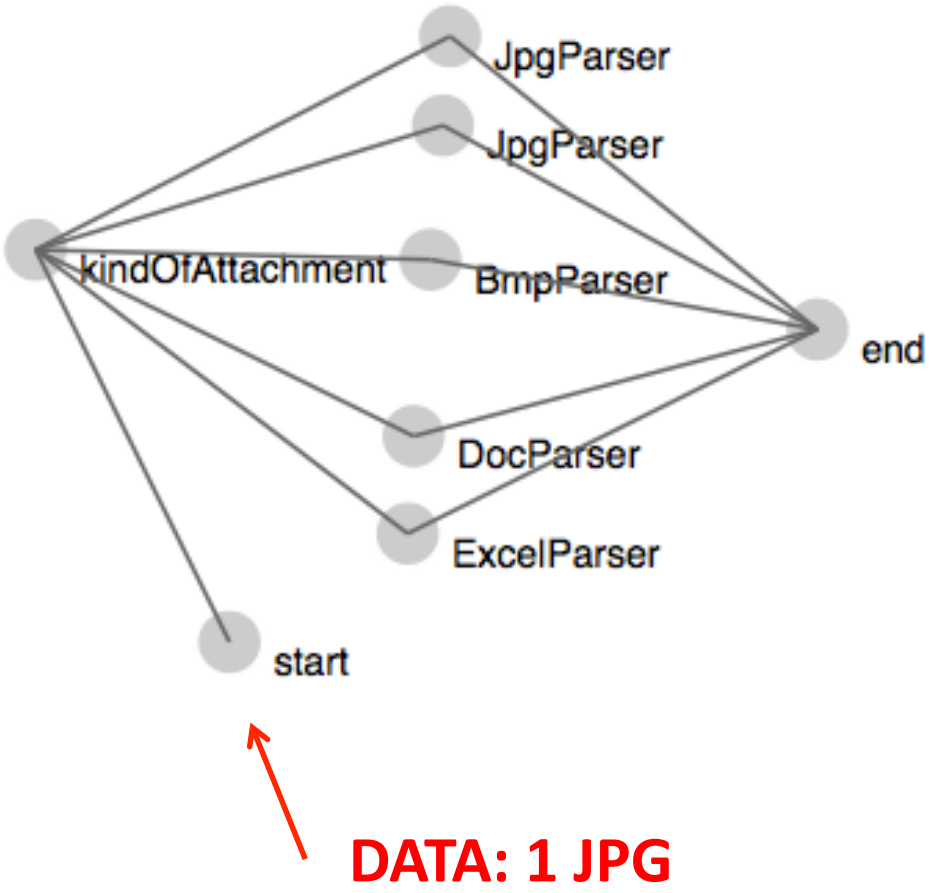
The use of work flow topology observables in a Security Autonomous Response Network

by Adriaan Dens

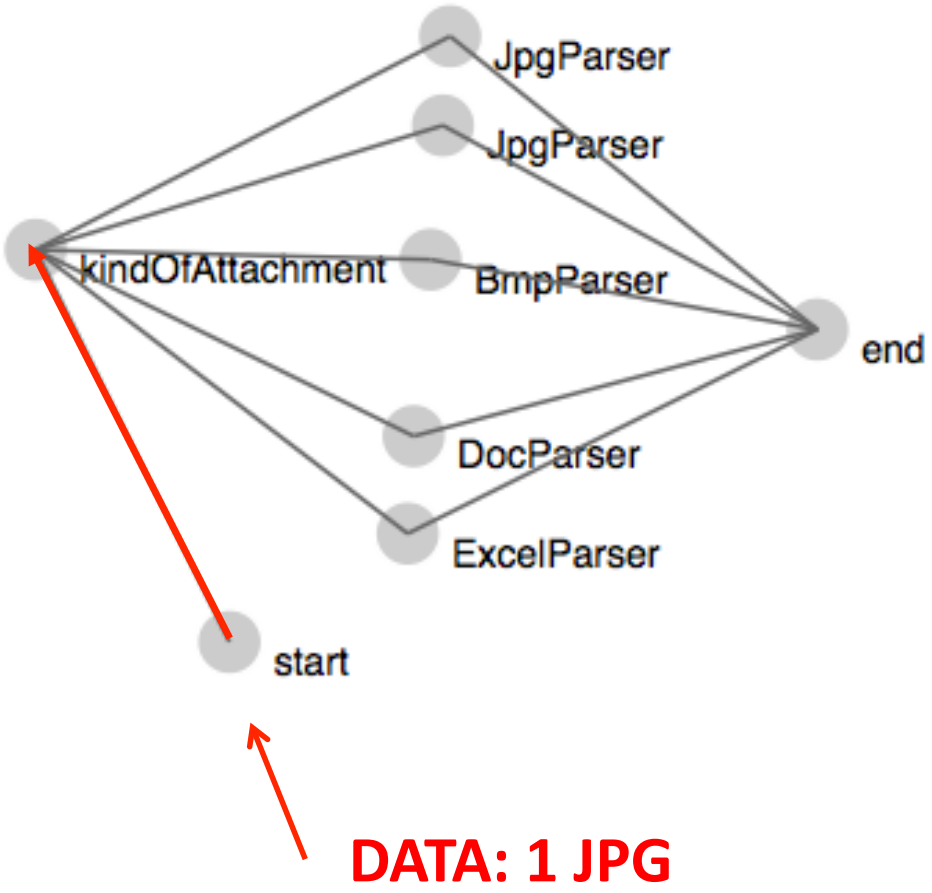
Supervised by Prof. Dr. Robert Meijer and Ir. Marc Makkes

The use of work flow topology observables in a Security Autonomous Response Network

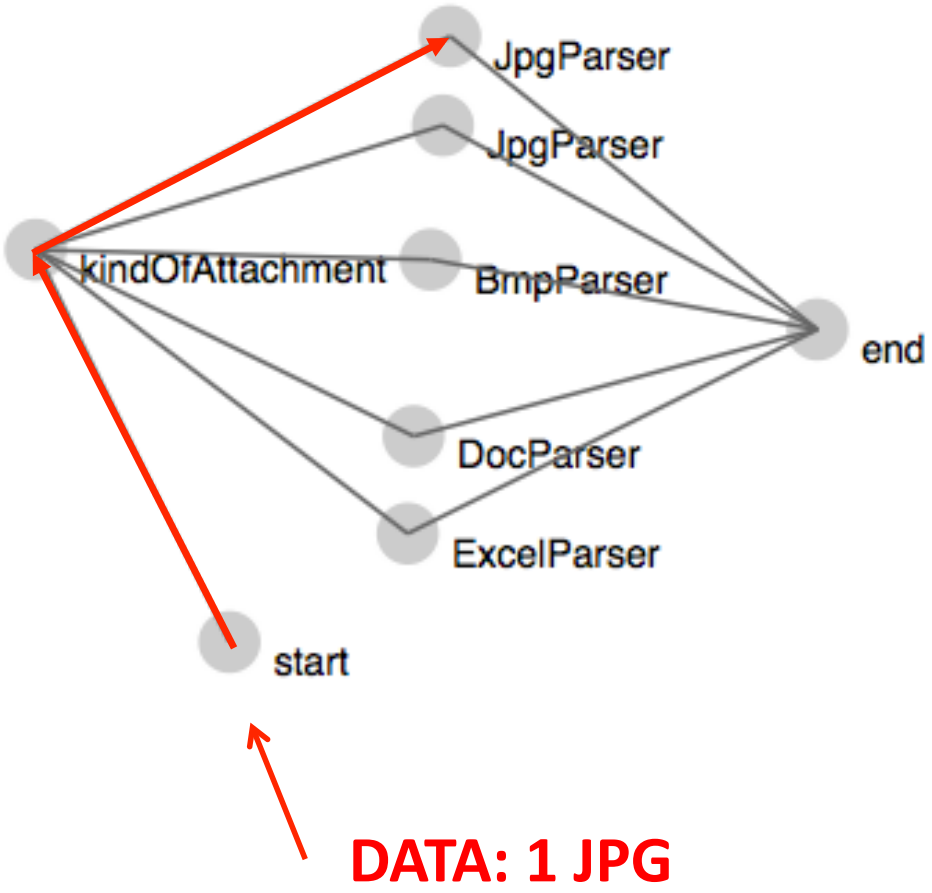
Work flow topologies



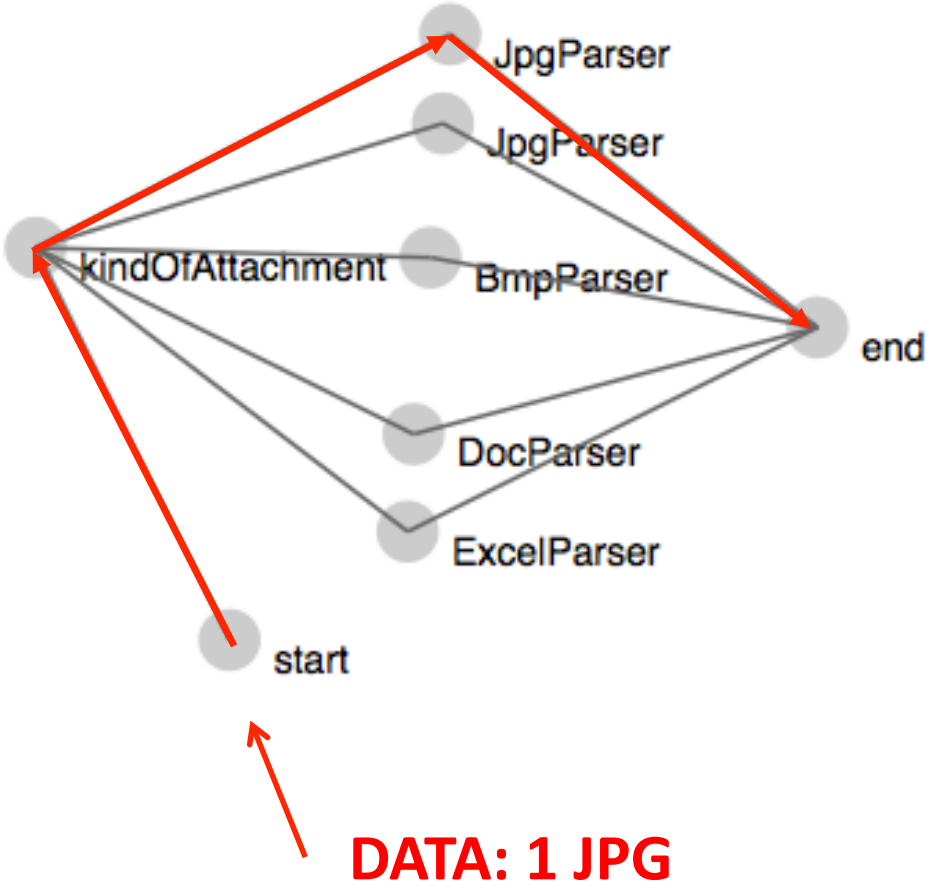
Work flow topologies



Work flow topologies



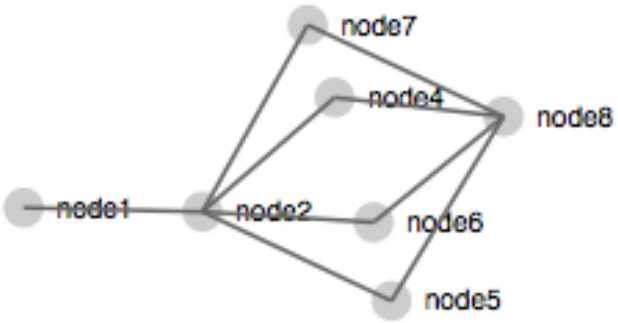
Work flow topologies

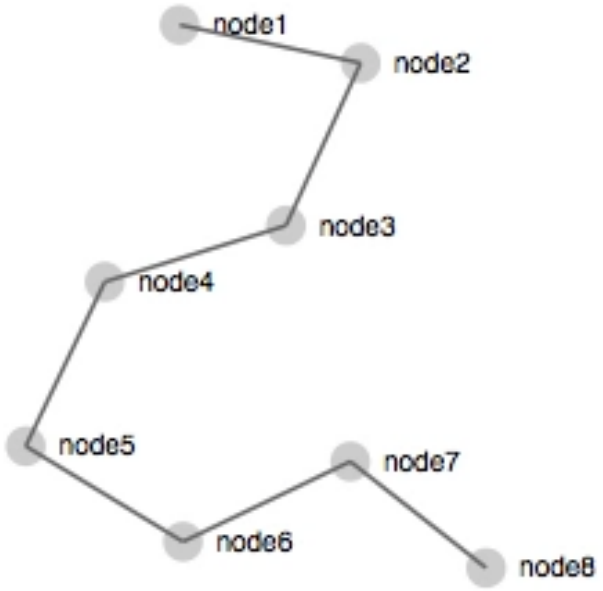


The use of work flow topology observables in a Security Autonomous Response Network

Observables of work flow topologies

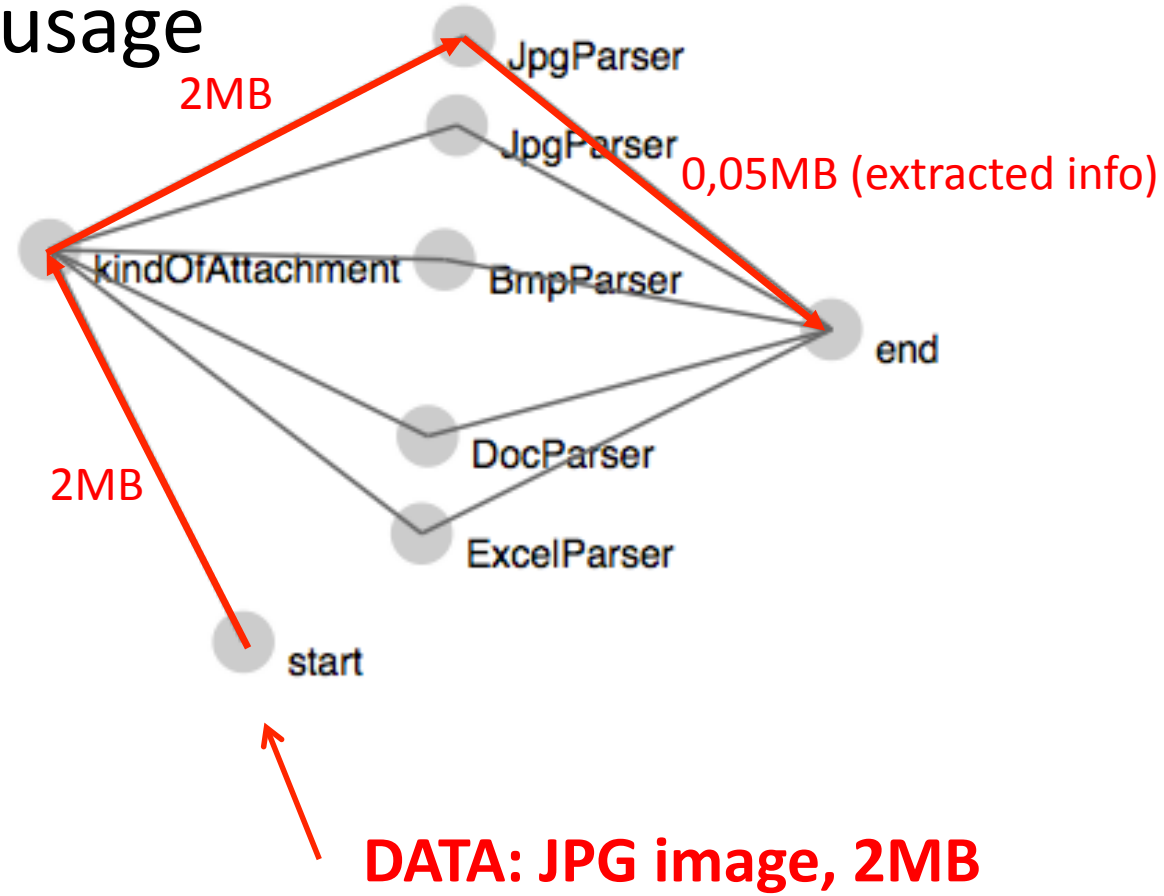
- The topology





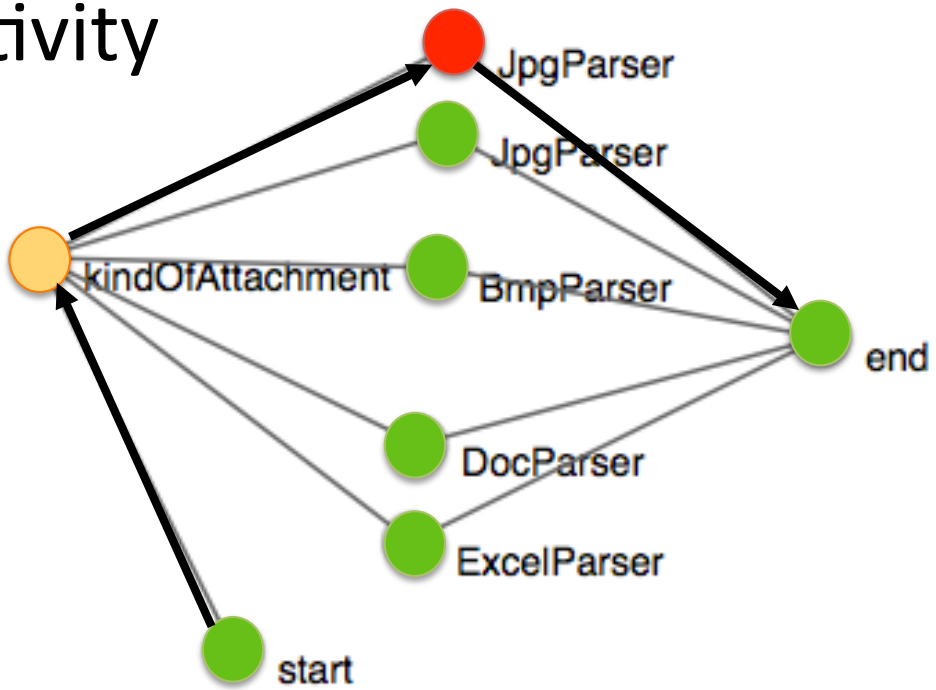
Observables of work flow topologies

- The link usage



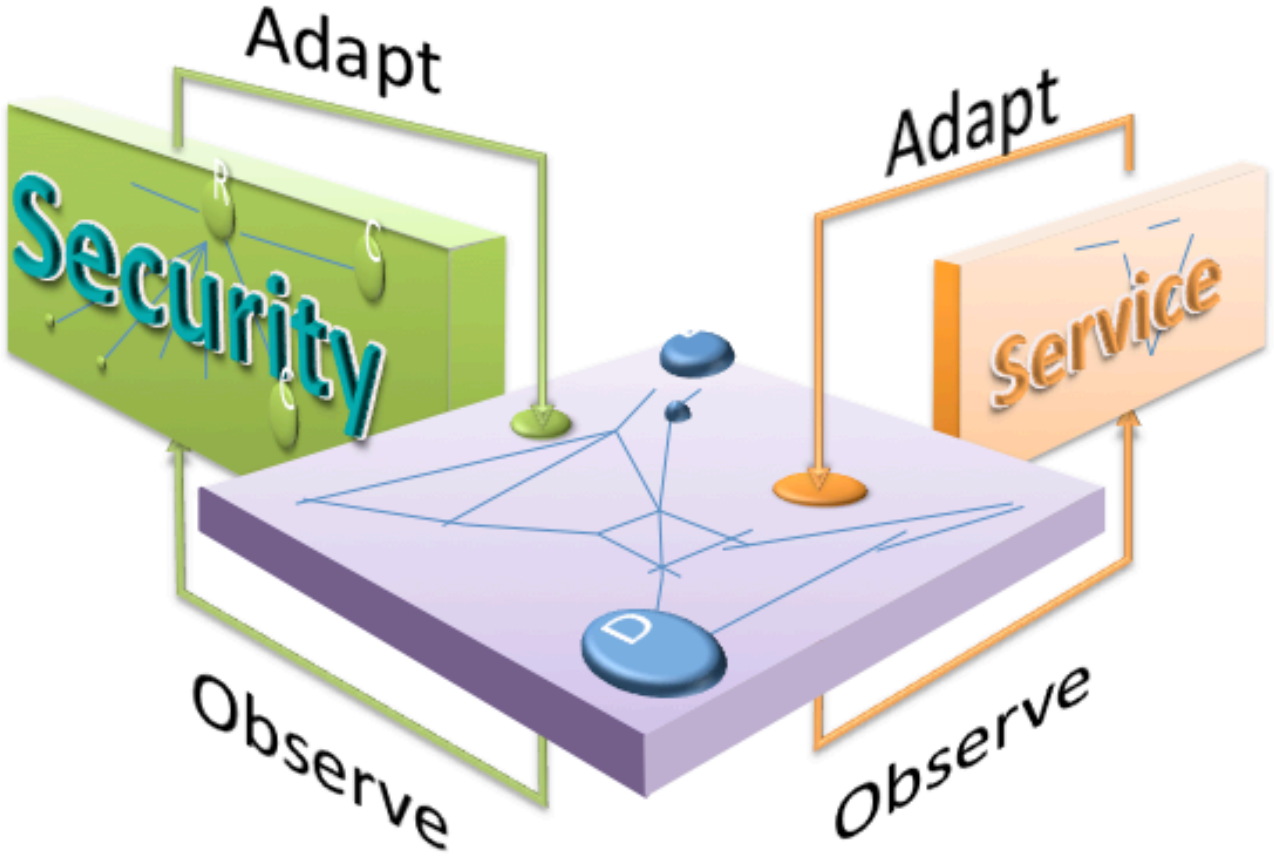
Observables of work flow topologies

- Node activity



DATA: JPG image, 2MB

The use of work flow topology observables in a Security Autonomous Response Network


Adapt
Security
R
C
Adapt
Service
D
Observe
Observe

Research Question

How can observables of software controlled work flow topologies be used in Security Autonomous Response networks?

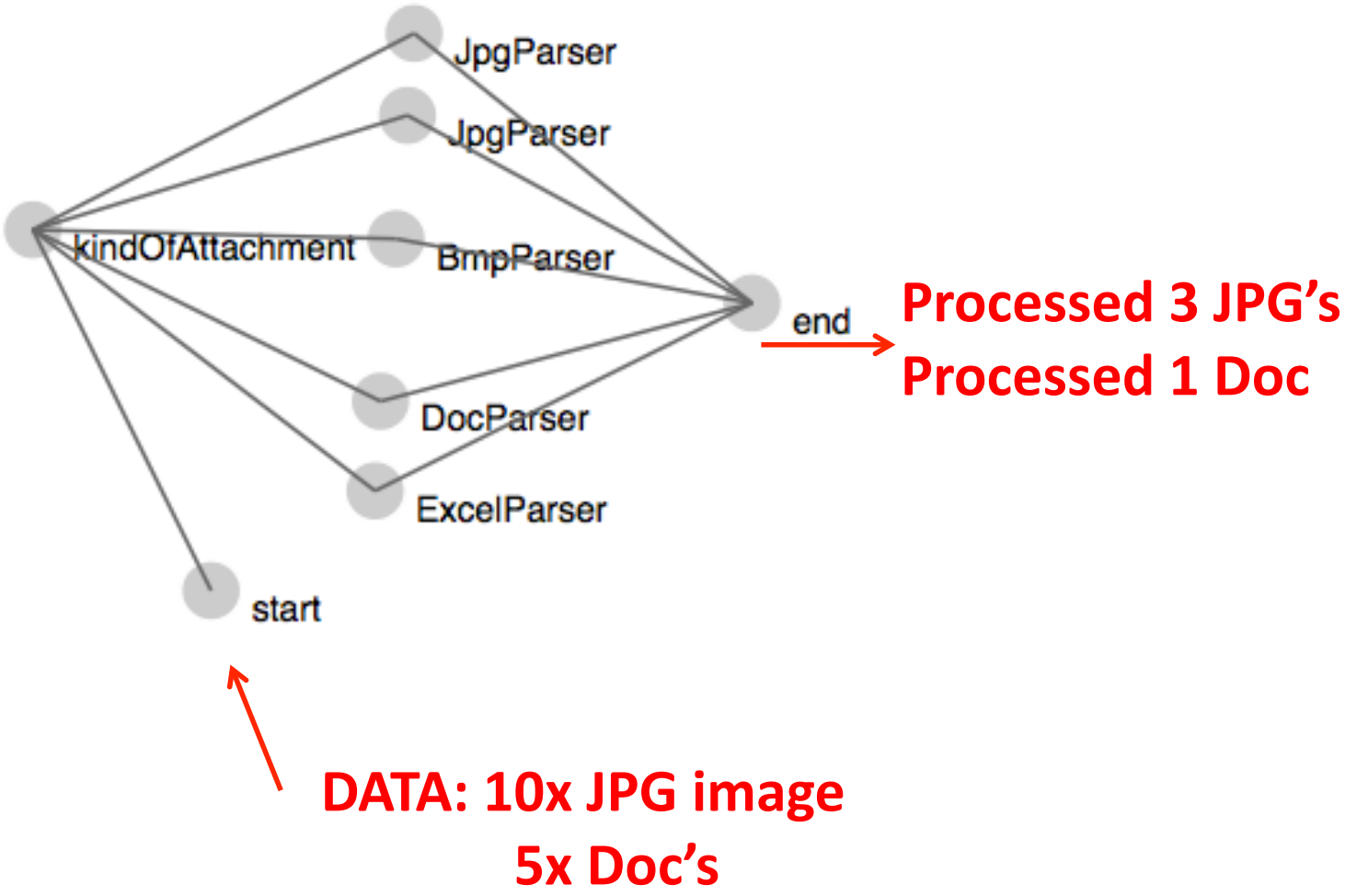
Relation between amount of (meta)data and things you can do

- Controlloop needs to know about topology
- Machine learning / benchmarking / …
- Lots of (meta)data => many things to control/ check

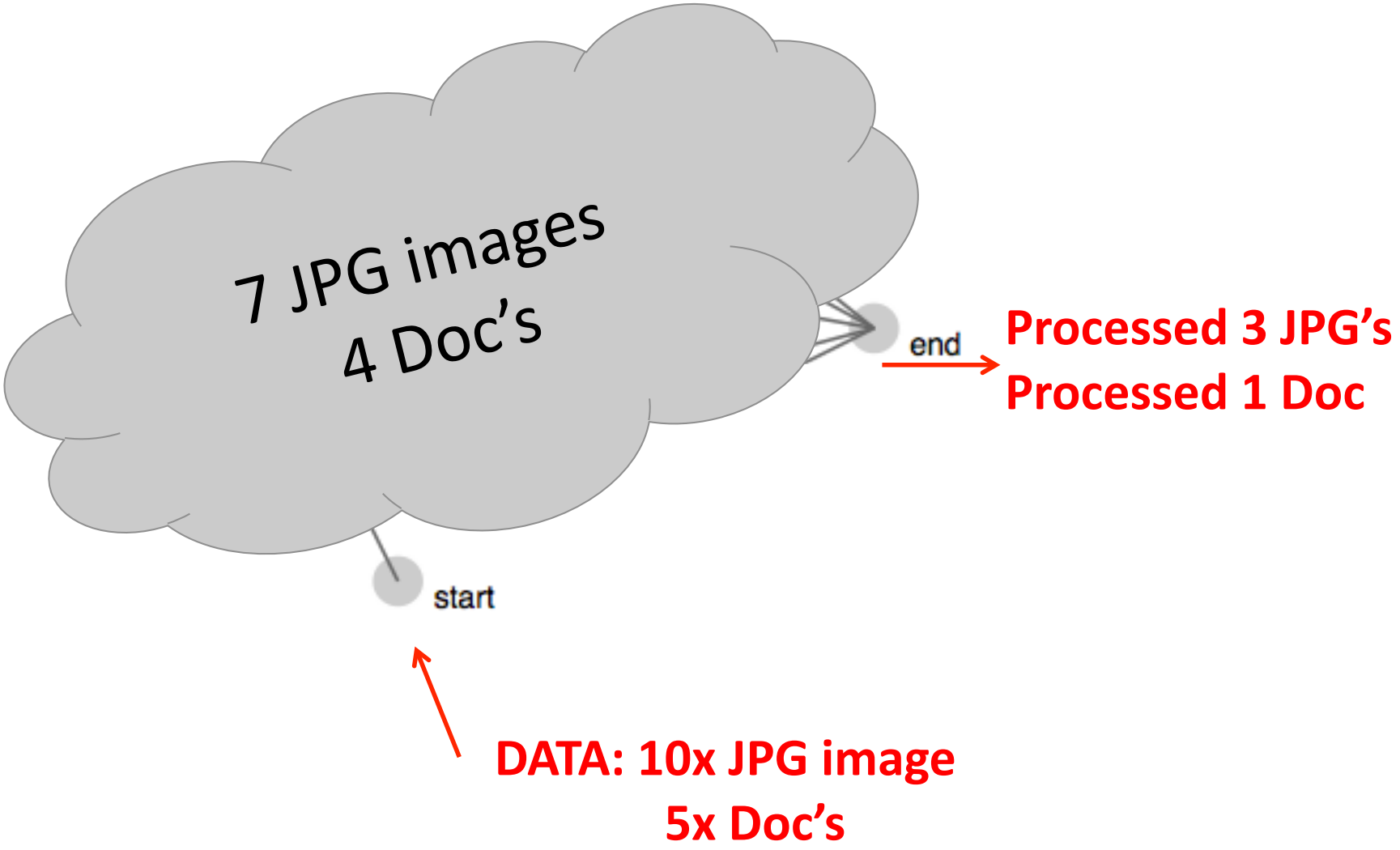
Observables & security

Data residing in the network tells something about the expected observables.

Observables & security



Observables & security



Observables & security

- Node activity: estimate activity…
- Link load: estimate link load…
- Topology: estimate topology…

… given data in the network

and compare to actual values of observables.

Estimating node activity and link load

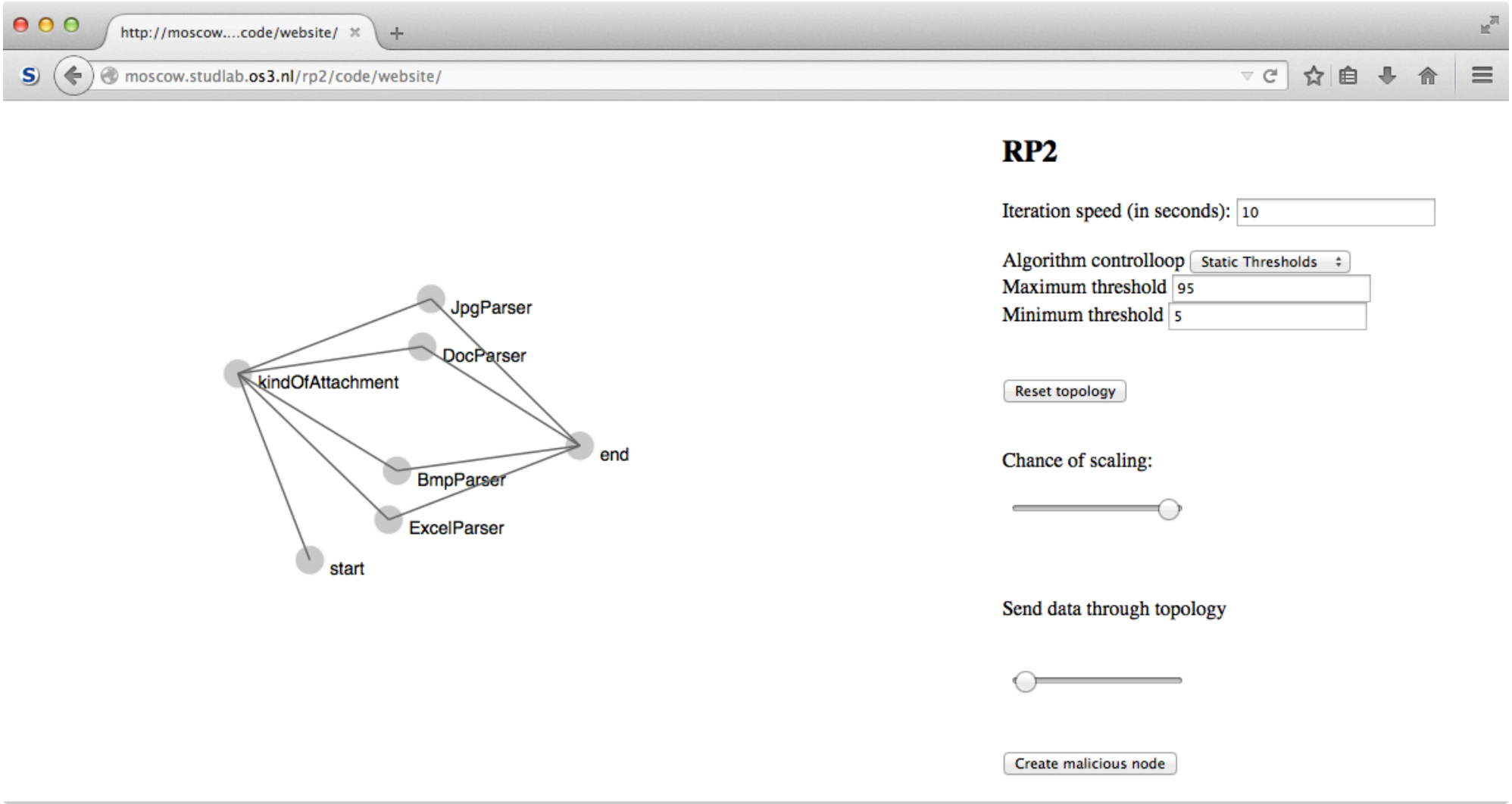
- Probability of data hitting a function
- Probability of data hitting a node
- Probability of data being in the node when sampling

Link load follows same intuition

Calculation of topology

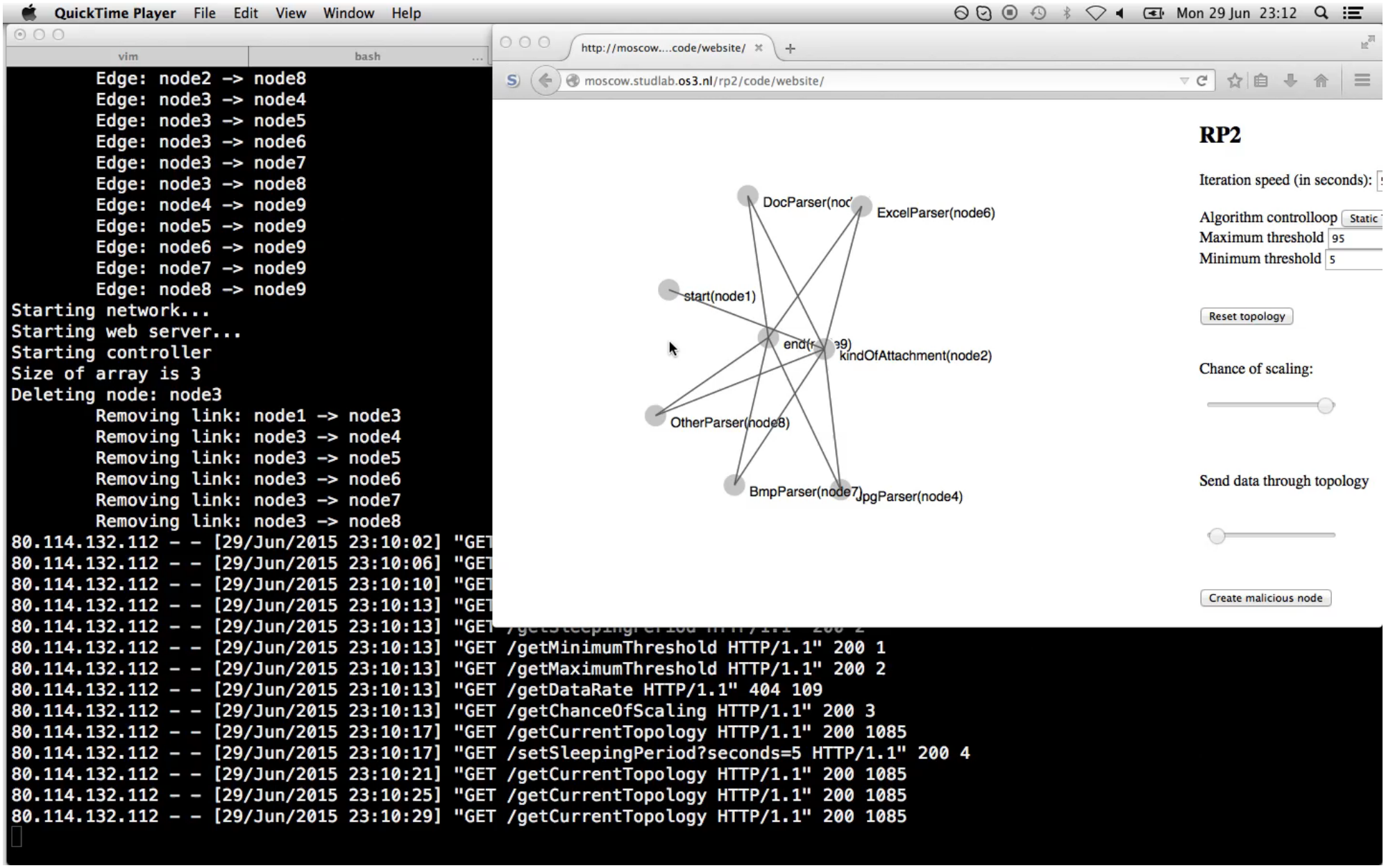
- Calculate number of nodes per function as upperbound: 1 <= real value <= upperbound
- Scale down the topology => compare with original topology

Web interface of Proof of Concept



Proof of Concept

- Developed control API for mininet:
  - But too slow for big networks
  - You cannot dynamically add hosts to mininet
- Therefore, pure simulation which uses same APIs
- PoC uses CPU load as node activity parameter


QuickTime Player File Edit View Window Help
Mon 29 Jun 23:12
vim
bash
Edge: node2 -> node8
Edge: node3 -> node4
Edge: node3 -> node5
Edge: node3 -> node6
Edge: node3 -> node7
Edge: node3 -> node8
Edge: node4 -> node9
Edge: node5 -> node9
Edge: node6 -> node9
Edge: node7 -> node9
Edge: node8 -> node9
Starting network...
Starting web server...
Starting controller
Size of array is 3
Deleting node: node3
Removing link: node1 -> node3
Removing link: node3 -> node4
Removing link: node3 -> node5
Removing link: node3 -> node6
Removing link: node3 -> node7
Removing link: node3 -> node8
80.114.132.112 - - [29/Jun/2015 23:10:02] "GET
80.114.132.112 - - [29/Jun/2015 23:10:06] "GET
80.114.132.112 - - [29/Jun/2015 23:10:10] "GET
80.114.132.112 - - [29/Jun/2015 23:10:13] "GET
80.114.132.112 - - [29/Jun/2015 23:10:13] "GET
80.114.132.112 - - [29/Jun/2015 23:10:13] "GET /getMinimumThreshold HTTP/1.1" 200 1
80.114.132.112 - - [29/Jun/2015 23:10:13] "GET /getMaximumThreshold HTTP/1.1" 200 2
80.114.132.112 - - [29/Jun/2015 23:10:13] "GET /getDataRate HTTP/1.1" 404 109
80.114.132.112 - - [29/Jun/2015 23:10:13] "GET /getChanceOfScaling HTTP/1.1" 200 3
80.114.132.112 - - [29/Jun/2015 23:10:17] "GET /getCurrentTopology HTTP/1.1" 200 1085
80.114.132.112 - - [29/Jun/2015 23:10:17] "GET /setSleepingPeriod?seconds=5 HTTP/1.1" 200 4
80.114.132.112 - - [29/Jun/2015 23:10:21] "GET /getCurrentTopology HTTP/1.1" 200 1085
80.114.132.112 - - [29/Jun/2015 23:10:25] "GET /getCurrentTopology HTTP/1.1" 200 1085
80.114.132.112 - - [29/Jun/2015 23:10:29] "GET /getCurrentTopology HTTP/1.1" 200 1085
http://moscow....code/website/
moscow.studlab.os3.nl/rp2/code/website/
DocParser(noc
ExcelParser(node6)
start(node1)
end(r
9)
kindOfAttachment(node2)
OtherParser(node8)
BmpParser(node7)
JpgParser(node4)
RP2
Iteration speed (in seconds):
Algorithm controlloop
Static
Maximum threshold
95
Minimum threshold
5
Reset topology
Chance of scaling:
Send data through topology
Create malicious node

Autonomous response

- Kill node
- Ignore node / Send fake data
- Extra monitoring
- Reprovisioning
- SDN flow rules

Conclusion

- Observables of work flow topologies can be used
  - By using metadata from the topology
  - Relation between knowledge of data and things you can do
  - More testing of equations is needed (finetuning)

Questions?