TECHNISCHE UNIVERSITÄT DRESDEN

Netzwerkkodierung in Theorie und Praxis

Praktische Anwendungen der Netzwerkkodierung

Professor Dr.-Ing. Dr. h.c. Frank H.P. Fitzek

M.Sc. Juan Cabrera

Deutsche Telekom Chair of Communication Networks (ComNets)

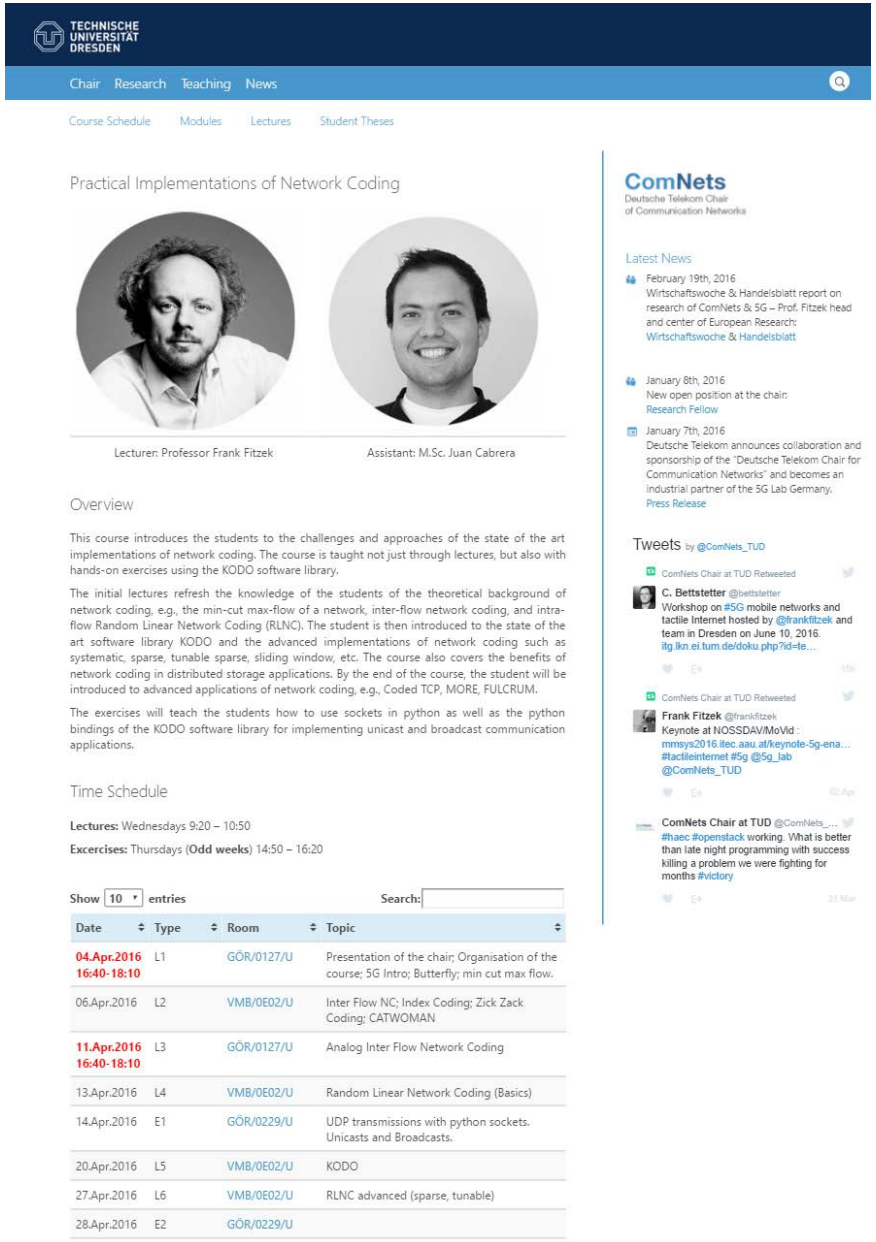
Netzwerkkodierungstheorie

Professor Dr.-Ing. Eduard Jorswieck

Dipl.-Ing. Johannes Richter

Theoretische Nachrichtentechnik

Lecture / Exercise Dates - tinyurl.com/zooafld



- Here all information for the lecture and the exercise can be found.
- Slides
- Links
  - Steinwurf
  - Python
  - KODOMARK (google play)

Please check every week!

TECHNISCHE UNIVERSITÄT DRESDEN

KODO: The network coding software library

Lecture 5

You need to get an GitHub Account

STEINWURF RESEARCH LICENSE

- Please read the license agreement
- "Research and Educational Use" means use for non-profit, non-commercial research and educational purposes only. Commercial use, including use for internal business purposes, is not permitted under this license.
- If you agree (and only if you agree), then ask for license

Kodo

- Random linear network coding library – Kodo
- Research license for free
- Use your TUD email and you will get access – add “participating lecture of Fitzek/Jorswieck at TUD”
- Use your TUD email account
- You need a github account

http://steinwurf.com/license/

steinwurf — Products | Technology | Development | About

License Request

Kodo is available under the following licenses:

- Research license: The license for non-commercial research and teaching, which can be found here.
- Training license: Time-limited license for the use of Kodo in connection with official Steinwurf trainings and webinars.
- Commercial Evaluation license: Time-limited license for the internal evaluation and testing of Kodo.
- Supported Commercial Evaluation license: If you need longer internal evaluation, support or additional software components, we offer an internal commercial evaluation and test license, at a one time fee which is fully refundable in any future Commercial license purchase(s). Contact us for details.
- Commercial license: A royalty-based license, which enables you to build and sell commercial products using Kodo. Contact us for details.

Obtain a Kodo License

Submit a request for a Kodo License

* Required

Full name *

Please provide your FULL name!

Affiliation *

University or Company.

Affiliation email *

Please provide your UNIVERSITY or COMPANY email address (you will receive an email to CONFIRM this address). Public email providers are not supported.

Invalid or unsupported email address

Country *

Github user *

A VALID Github username is required to provide access to the software (please REGISTER on github.com if you don't have an account)

Type of license *

Please select the appropriate license based on the descriptions above

- Research
- Training
- Commercial Evaluation
- Commercial

Description of use *

Please provide a paragraph about your planned use of Kodo: What is the goal of your research? What kind of system you plan to implement or simulate?

Must be at least 50 characters and it cannot include newlines.

Submit

FAQ

What is research use?

Please read the Steinwurf Research License first. Briefly, research use is research and teaching conducted at educational institutions and similar with no commercial purpose.

If I use Kodo for research, does Steinwurf hold any rights to my work?

No.

My question is not answered here!

Please contact us with your question(s).

STEINWURF APS

SEARCH THIS WEBSITE...

What we want to install!

kodo

Main C++ software library

kodo-basic-simulations

Simulation environment for wireless mesh networks (simple)

kodo-python

User API of kodo for Python

fifi-python

Finite field calculus for Python

What we want to install!

kodo

Main C++ software library

kodo-basic-sim

kodo-simulations-python

Simulation environment for wireless mesh networks (simple)

kodo-python

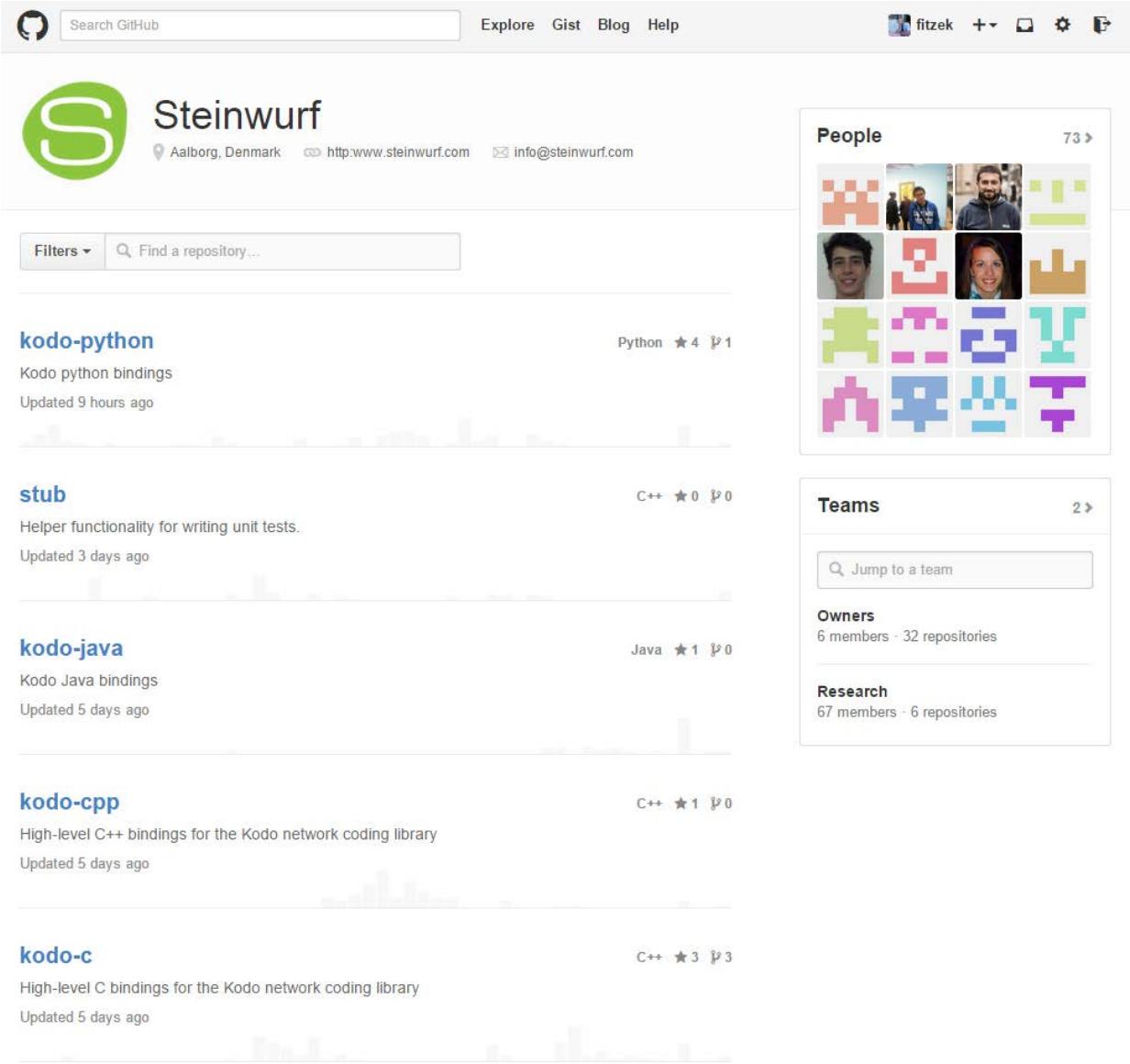
User API of kodo for Python

fifi-python

Finite field calculus for Python

Installing

- Where on github can I find the sources?
- https://github.com/steinwurf
- Following examples for UBUNTU



Prerequisite

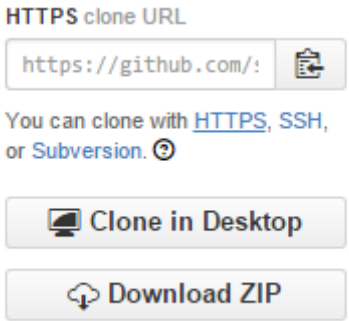
kodo

- sudo apt-get install g++ python git-core

Installing kodo using git with HTTPS

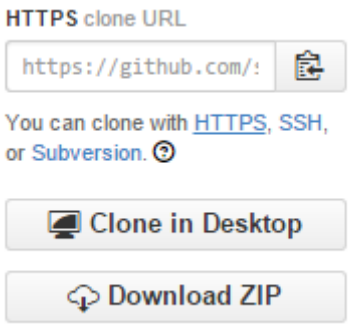
- git clone https://github.com/steinwurf/kodo.git
  - Needs your github user name and password
- cd kodo
- python waf configure
  - Takes some time
- python waf build
  - Takes some time
- cd ..

kodo



Installing kodo using git

- git clone git@github.com:steinwurf/kodo.git
  - Then your git needs to know your user name and password apriori
- cd kodo
- python waf configure
  - Takes some time
- python waf build
  - Takes some time
- cd ..



HINT:
On virtual machine with little memory, please use *python waf build –j1*

HINT:
https://help.github.com/articles/set-up-git/

Prerequisite

kodo-python

- sudo apt-get update
- sudo apt-get install python git build-essential libpython-dev

Installing kodo-python

- git clone https://github.com/steinwurf/kodo-python.git
  - Needs your github user name and password
- cd kodo-python
- python waf configure
  - Takes some time
- python waf build
  - Takes some time
- cd ..

kodo-python

Installing kodo-python

kodo-python

- Try to start python and write “import kodo”
- This should lead to an error as the module kodo is not known
- In order to make the kodo module known to python you need to add to your .bashrc

```
export PYTHONPATH=/XXXXX/kodo-python/build/linux/src/kodo_python
```

- If you start a new shell or resource the .bashrc and call import kodo now in python it should work
- In order to see the graphical output you need pygame

```
sudo get-apt install python-pygame
```

Installing fifi-python

- git clone https://github.com/steinwurf/fifi-python.git
  - Needs your github user name and password
- cd fifi-python
- python waf configure
  - Takes some time
- python waf build
  - Takes some time
- cd ..

fifi-python

Installing fifi-python

fifi-python

- Try to start python and write “import fifi”
- This should lead to an error as the module fifi is not known
- In order to make the fifi module known to python you need to add to your .bashrc

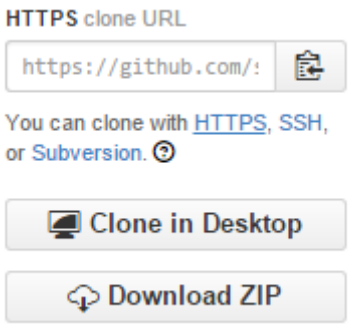
  export PYTHONPATH=/XXXXX/fifi-python/build/linux/src/fifi-python***:$PYTHONPATH***

- If you start a new shell or resource the .bashrc and call import fifi now in python it should work

Installing kodo-basic-simulations

- git clone https://github.com/steinwurf/kodo-basic-simulations
  - Needs your github user name and password
- cd kodo-basic-simulations
- python waf configure
  - Takes some time
- python waf build
  - Takes some time
- ./build/linux/relay/relay_simulations/relay_simulations
  - Test if the simulator works, will come back with a bunch of numbers
- cd ..

kodo-basic-simulations



Downloading kodo-basic-simulations

kodo-basic-simulations

- In case you just want to have quick access to the simulator, just follow the instruction below: http://176.28.49.184:12344/bin/
- There is platform support for
  - iOS
  - Windows
  - Several Linux versions
    - Android
    - Raspian
    - Ubuntu/debian
- Just pick the right one!

Want an executable?

We use a buildbot to build and test the code on a number of different platforms. If you just want an executable to play with our buildbots upload theirs here:

http://176.28.49.184:12344/bin

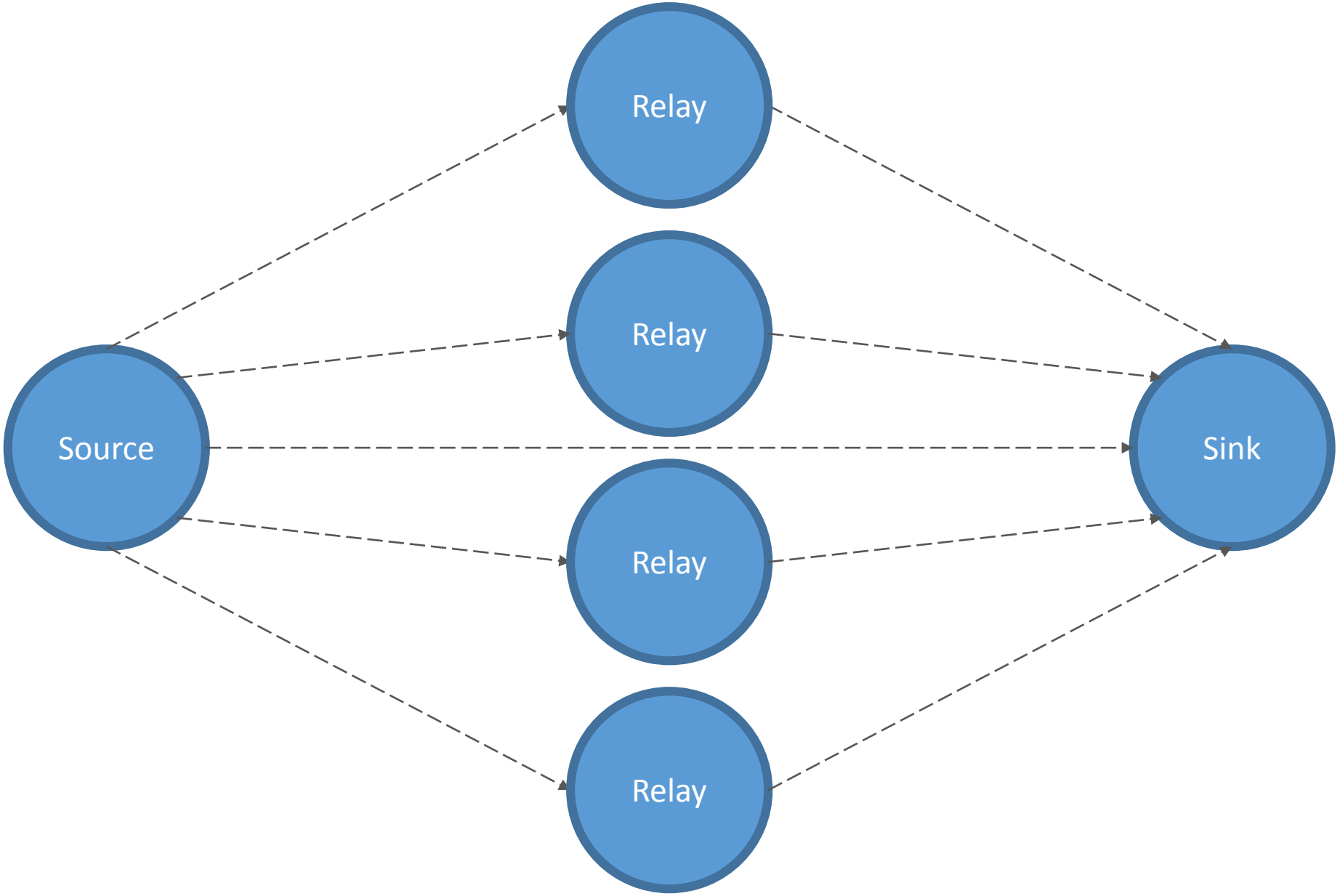
Look in the folder `kodo-basic-simulations`.

Did it all work out?

Simulator

kodo-basic-simulations

What the simulator does!



Medium Access Control

- The medium access control (MAC) is very simple
- The source will send first, then relay_0, then relay_1, etc
- All nodes will get the same capacity if they want (beside the sink, that obviously does not need anything!)
- IEEE802.11's policy is also to give all nodes the same capacity
- In the source code of kodo-basic-simulator it is very easy to change that policy if needed

Starting the simulator

- First start the simulator with “--help”
- Important parameters
  - runs
  - symbols vs symbol_size
  - Error probability
  - relays

```
C:\Users\frank\Downloads>"relay_simulations (1).exe" --help
Gauge:
  --help                                produce help message
  --result_filter arg                   Filter which results should be stored
                                        for example ./benchmark
                                        --result_filter=time multiple filters
                                        can be a comma separated list of filters
                                        e.g. --result_filter=time throughput
  --gauge_filter arg                    Filter which benchmarks to run based on
                                        their name for example ./benchmark
                                        --gauge_filter=MyTest.* multiple filters
                                        can also be specified e.g.
                                        --gauge_filter=MyTest.one MyTest.two
  --runs arg                            Sets the number of runs to complete.
                                        Overrides the settings specified in the
                                        benchmark ex. --runs=50
  --add_column arg                      Add a column to the test results, this
                                        can be use to add custom information to
                                        the result files ./benchmark
                                        --add_column cpu=i7 "date=Monday 1st
                                        June 2021"

  --symbols arg (=32)                   Set symbols
  --symbol_size arg (=1400)             Set symbols size
  --error_source_sink arg (=0.5)        Error source to sink
  --error_source_relay arg (=0.5)       Error source to relay
  --error_relay_sink arg (=0.5)         Error relay to sink
  --relays arg (=1)                     Relays
  --source_systematic arg (=1)          Whether the source is systematic or not
                                        --systematic=1 turns on systematic
                                        source. Systematic means that all
                                        packets in a generation are sent first
                                        once without coding. After sending
                                        everything once coding starts.
  --relay_transmit_on_receive arg (=1) Set true if the relay(s) should transmit
                                        in every tick or when a packet is
                                        received from the source
  --recode arg (=1)                     Set true if the relay(s) should recode
                                        packets
  --pyfile arg (=out.py)                Set the output name of the python
                                        printer
  --csvfile arg (=out.csv)              Set the output name of the csv printer
  --csvseperator arg (=,)               Set the value separator for the csv file
                                        writer


C:\Users\frank\Downloads>
```

Output for the binary case

This is the result for the binary field size!

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

Only 10 runs!

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

CONFIG: Tells you all parameters again, but good for cross checking it later!

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[    TIME  ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

TIME: How long did it take to run the simulation in ms.

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

There are three links and the channel reports for those that are ok and those that have been dropped.

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

Innovative versus linear dependent!

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

Where the sink got the packets from!

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary case

How many packets the source has sent out!

```
[ RUN      ] Relay.binary (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systen
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 6.456000 milliseconds
[   RESULT ] channel_relay0_to_sink_dropped
[          ]    Average: 13.000000 packets
[          ]        Max: 19.000000 packets (+6.000000 packets / +46.153846 %)
[          ]        Min: 8.000000 packets (-5.000000 packets / -38.461538 %)
[   RESULT ] channel_relay0_to_sink_ok
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]    Average: 24.000000 packets
[          ]        Max: 31.000000 packets (+7.000000 packets / +29.166667 %)
[          ]        Min: 20.000000 packets (-4.000000 packets / -16.666667 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]    Average: 24.600000 packets
[          ]        Max: 35.000000 packets (+10.400000 packets / +42.276423 %)
[          ]        Min: 19.000000 packets (-5.600000 packets / -22.764228 %)
[   RESULT ] channel_source_to_sink_ok
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] relay0_innovative_from_source
[          ]    Average: 24.900000 packets
[          ]        Max: 31.000000 packets (+6.100000 packets / +24.497992 %)
[          ]        Min: 21.000000 packets (-3.900000 packets / -15.662651 %)
[   RESULT ] sink_innovative_from_relay0
[          ]    Average: 8.400000 packets
[          ]        Max: 11.000000 packets (+2.600000 packets / +30.952381 %)
[          ]        Min: 5.000000 packets (-3.400000 packets / -40.476190 %)
[   RESULT ] sink_innovative_from_source
[          ]    Average: 23.600000 packets
[          ]        Max: 27.000000 packets (+3.400000 packets / +14.406780 %)
[          ]        Min: 21.000000 packets (-2.600000 packets / -11.016949 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]    Average: 3.500000 packets
[          ]        Max: 7.000000 packets (+3.500000 packets / +100.000000 %)
[          ]        Min: 0.000000 packets (-3.500000 packets / -100.000000 %)
[   RESULT ] sink_linear_dept_from_source
[          ]    Average: 0.700000 packets
[          ]        Max: 2.000000 packets (+1.300000 packets / +185.714286 %)
[          ]        Min: 0.000000 packets (-0.700000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]    Average: 11.900000 packets
[          ]        Max: 18.000000 packets (+6.100000 packets / +51.260504 %)
[          ]        Min: 7.000000 packets (-4.900000 packets / -41.176471 %)
[   RESULT ] sink_receive_from_source
[          ]    Average: 24.300000 packets
[          ]        Max: 27.000000 packets (+2.700000 packets / +11.111111 %)
[          ]        Min: 21.000000 packets (-3.300000 packets / -13.580247 %)
[   RESULT ] source_sent
[          ]    Average: 48.900000 packets
[          ]        Max: 59.000000 packets (+10.100000 packets / +20.654397 %)
[          ]        Min: 43.000000 packets (-5.900000 packets / -12.065440 %)
[----------]
```

Output for the binary8 case

```
[----------]
[ RUN      ] Relay.binary8 (10 runs / 1.000000 iteration)
[  CONFIG  ] error_relay_sink=0.500000, error_source_relay=0.500000, error_source_sink=0.500000, recode=1, relay_transmit_on_receive=1, relays=1, source_systema
tic=1, symbol_size=1400, symbols=32
[   TIME   ] 11.[illegible]2400[illegible] seconds
[   RESULT ] channel_[illegible]_sink_dropped
[          ]     Average: [illegible]400000 packets
[          ]         Max: [illegible]000000 packets (+3.600000 packets / +29.032258 %)
[          ]         Min: [illegible]00000 packets (-3.400000 packets / -27.419355 %)
[   RESULT ] channel_r[illegible]to_sink_ok
[          ]     Average: 12.200000 packets
[          ]         Max: 16.000000 packets (+3.800000 packets / +31.147541 %)
[          ]         Min: 9.000000 packets (-3.200000 packets / -26.229508 %)
[   RESULT ] channel_source_to_relay0_dropped
[          ]     Average: 21.400000 packets
[          ]         Max: 32.000000 packets (+10.600000 packets / +49.532710 %)
[          ]         Min: 15.000000 packets (-6.400000 packets / -29.906542 %)
[   RESULT ] channel_source_to_relay0_ok
[          ]     Average: 24.600000 packets
[          ]         Max: 28.000000 packets (+3.400000 packets / +13.821138 %)
[          ]         Min: 21.000000 packets (-3.600000 packets / -14.634146 %)
[   RESULT ] channel_source_to_sink_dropped
[          ]     Average: 23.700000 packets
[          ]         Max: 32.000000 packets (+8.300000 packets / +35.021097 %)
[          ]         Min: 16.000000 packets (-7.700000 packets / -32.489451 %)
[   RESULT ] channel_source_to_sink_ok
[          ]     Average: 22.300000 packets
[          ]         Max: 27.000000 packets (+4.700000 packets / +21.076233 %)
[          ]         Min: 16.000000 packets (-6.300000 packets / -28.251121 %)
[   RESULT ] relay0_innovative_from_source
[          ]     Average: 24.600000 packets
[          ]         Max: 28.000000 packets (+3.400000 packets / +13.821138 %)
[          ]         Min: 21.000000 packets (-3.600000 packets / -14.634146 %)
[   RESULT ] sink_innovative_from_relay0
[          ]     Average: 9.700000 packets
[          ]         Max: 16.000000 packets (+6.300000 packets / +64.948454 %)
[          ]         Min: 5.000000 packets (-4.700000 packets / -48.453608 %)
[   RESULT ] sink_innovative_from_source
[          ]     Average: 22.300000 packets
[          ]         Max: 27.000000 packets (+4.700000 packets / +21.076233 %)
[          ]         Min: 16.000000 packets (-6.300000 packets / -28.251121 %)
[   RESULT ] sink_linear_dept_from_relay0
[          ]     Average: 2.200000 packets
[          ]         Max: 6.000000 packets (+3.800000 packets / +172.727273 %)
[          ]         Min: 0.000000 packets (-2.200000 packets / -100.000000 %)
[   RESULT ] sink_receive_from_relay0
[          ]     Average: 12.200000 packets
[          ]         Max: 16.000000 packets (+3.800000 packets / +31.147541 %)
[          ]         Min: 9.000000 packets (-3.200000 packets / -26.229508 %)
[   RESULT ] sink_receive_from_source
[          ]     Average: 22.300000 packets
[          ]         Max: 27.000000 packets (+4.700000 packets / +21.076233 %)
[          ]         Min: 16.000000 packets (-6.300000 packets / -28.251121 %)
[   RESULT ] sink_waste_from_relay0
[          ]     Average: 0.300000 packets
[          ]         Max: 1.000000 packets (+0.700000 packets / +233.333333 %)
[          ]         Min: 0.000000 packets (-0.300000 packets / -100.000000 %)
[   RESULT ] source_sent
[          ]     Average: 46.000000 packets
[          ]         Max: 58.000000 packets (+12.000000 packets / +26.086957 %)
[          ]         Min: 41.000000 packets (-5.000000 packets / -10.869565 %)
[----------]
```

8

This is the result for the binary8 field size!

How do I change the paramters?

- Ofc you can change the parameter
- Call the simulator with *--parameter value* (two dashes)

simulator --parameter1 value --parameter2 value …

simulator --runs 1000

(will change the number of runs to 1000 the rest are default values)

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

How many generations are transmitted from source to sink?

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

How many packets form one generation?

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

How many bytes has an original packet?
(can be low for simulation purposes)

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

What is the error probability for the direct link?

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

What is the error probability from the source to ALL relays?

(if you want to change this have a look at the source code)

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

What is the error probability from ALL relays to the sink?

(if you want to change this have a look at the source code)

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

How many relays we have?

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

Do you want to use systematic coding?

How do I change the paramters?

- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

What is the transmitting policy for the simulator? Always transmit (0) or only if rank has increased (1)?

How do I change the paramters?

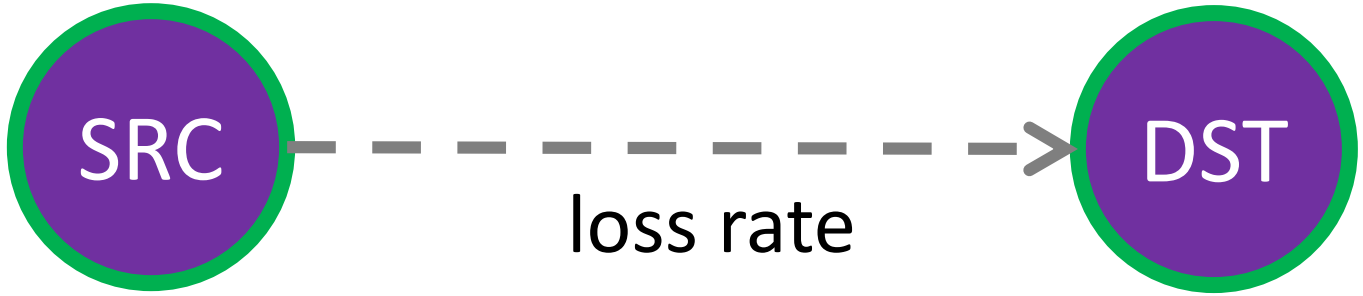
- Best is to make a small bash script to avoid errors

simulator --runs 1000 --symbols 16 --symbol_size 40 --error_source_sink 0.8 --error_source_relay 0.1 --error_relay_sink 0.2 --relays 10 --source_systematic 0 –relay_transmit_on_receive 0 --recode 1

Do you allow the relays to recode?
Of course, but just to see …

TASK 1

- In the first example we use a point to point communication system. Our performance metric will be the number of sent packets (Nc) from the source per original packet in order to guarantee the recovery of all M original packets.



TASK 1

- Start the program with the following parameters and report the number of packets send by the source per successful packet (*) and the number of linear dependent packets.

Parameter	Value
field	{binary|binary8|binary16}
source_systematic	{0|1}
symbols	{8|1024}
symbol_size	100
runs	100.000 for generation size 8 100 for generation size 1014
recode	1
relays	0
error_source_sink	{0|0.5}

- (*) KODO will give the absolute number of packets sent. This number should be divided by the generation size and the number of iterations to be able to have a fair comparison amongst different configurations (e.g., different packet size, generation size, field size, etc).

TASK 1

- What is the estimated number of sent packets in the error free case?
- What will change in the error-prone case?
- What will change if we switch on the systematic code?
- How will a larger generation size impact the results?
- Do you notice any delay in achieving the results for the different field sizes?

TASK 1: Error-free case 0%

Parameter			Value	
field	generation size	systematic	Number of packets sent per useful packet	Number of linearly dependent packets at the sink
binary	8	0		
binary 8	8	0		
binary 16	8	0		

Parameter			Value	
field	generation size	systematic	Number of packets sent per useful packet	Number of linear dependent packets at the sink
binary	8	1		
binary 8	8	1		
binary 16	8	1		

TASK 1: Error-prone case 50%

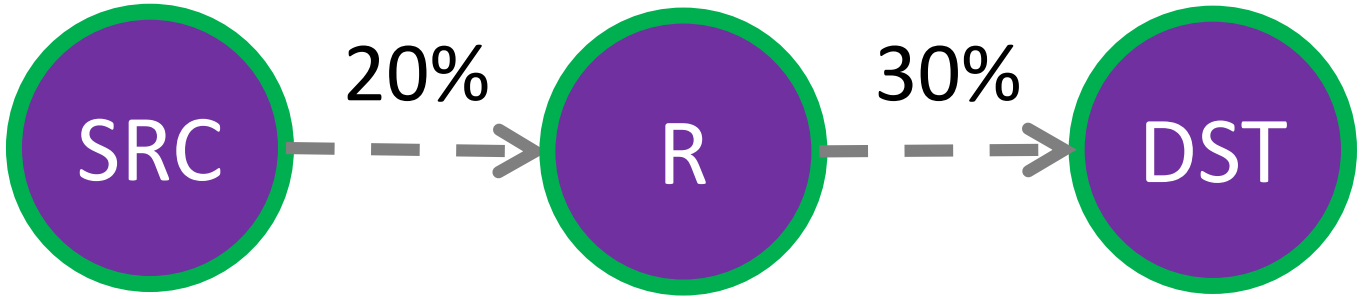
Parameter			Value	
field	generation size	systematic	Number of packets sent per useful packet	Number of linearly dependent packets at the sink
binary	8	0		
binary 8	8	0		
binary 16	8	0		

Parameter			Value	
field	generation size	systematic	Number of packets sent per useful packet	Number of linear dependent packets at the sink
binary	8	1		
binary 8	8	1		
binary 16	8	1		

TASK 1

Parameter			Value	
field	generation size	Systematic	Number of packets sent per useful packet	Number of linear dependent packets at the sink
binary	1024	1		
binary 8	1024	1		
binary 16	1024	1		
binary	1024	0		
Binary8	1024	0		
binary16	1024	0		

Task 2

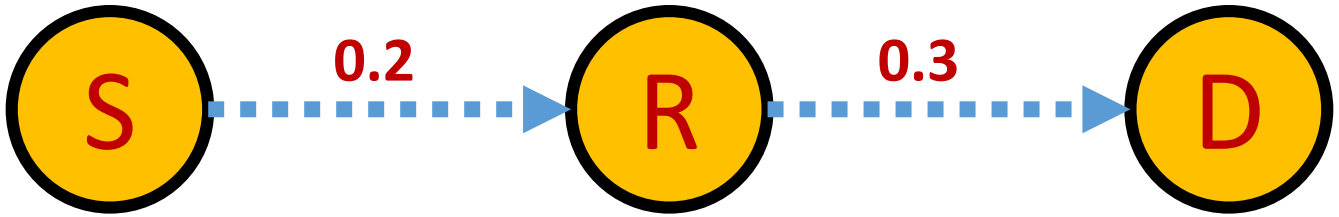


Parameter	Value
field	(binary|binary8|binary16)
source_systematic	0
symbols	1024
symbol_size	100
runs	CHOOSE
recode	{0|1}
relays	1
error_source_sink	1.0
error_source_relays	0.2
error_relays_sink	0.3

Task 2

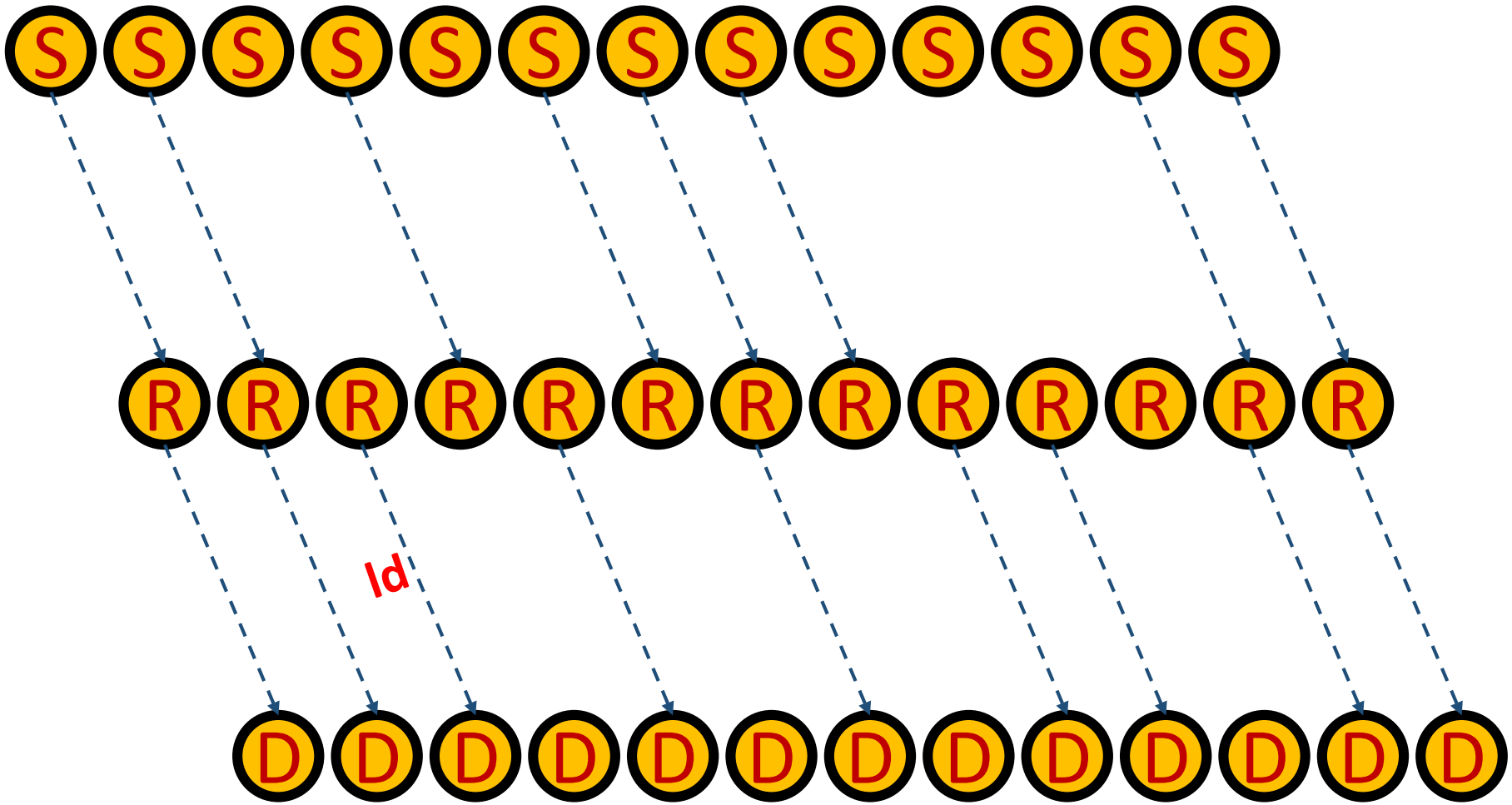
Parameter		Value
field	recode	Number of packets send by the source
binary	1	
binary 8	1	
binary 16	1	
any	0	

Task 2

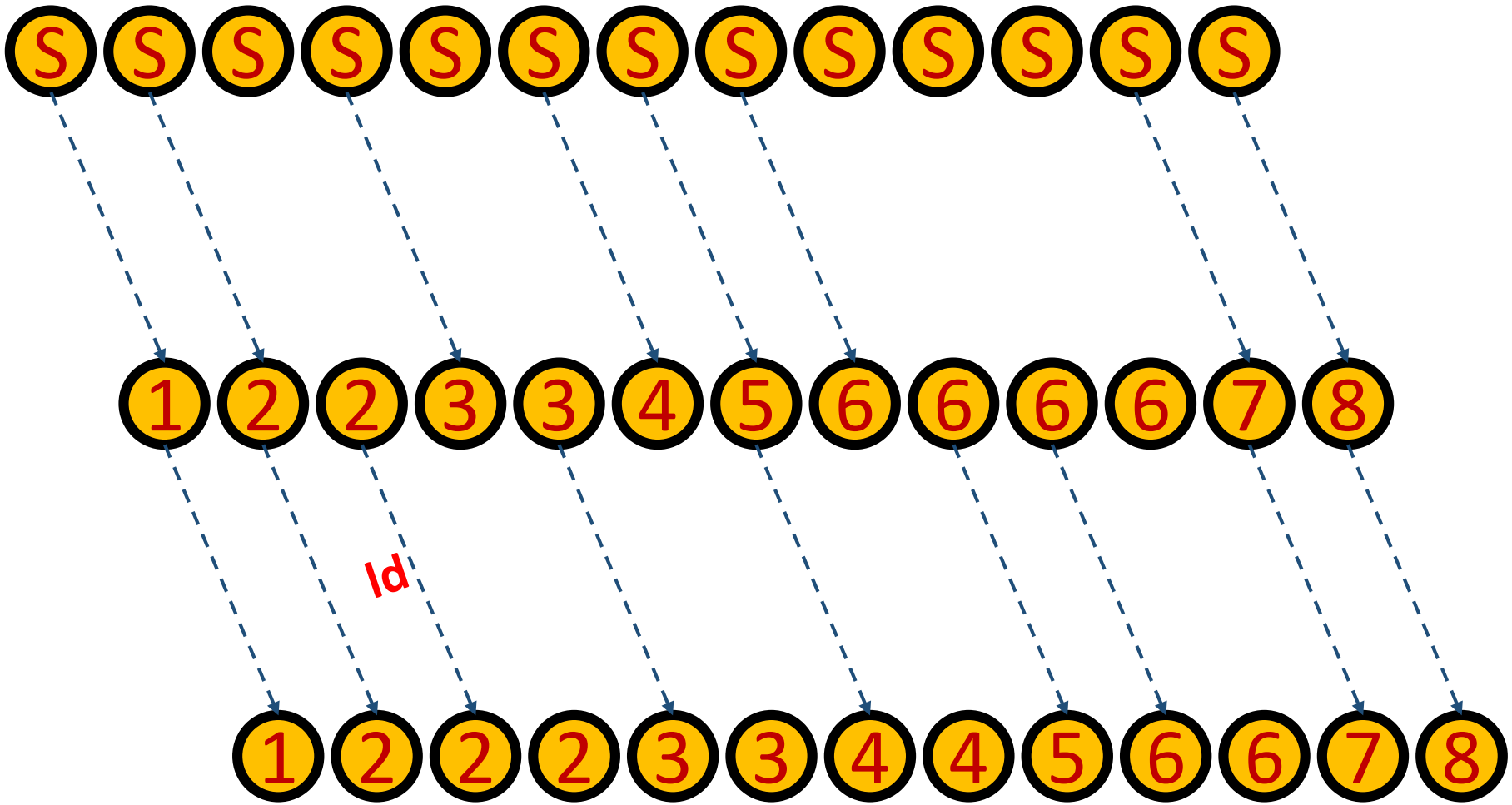


- Channel capacity 0.56 improved to 0.70

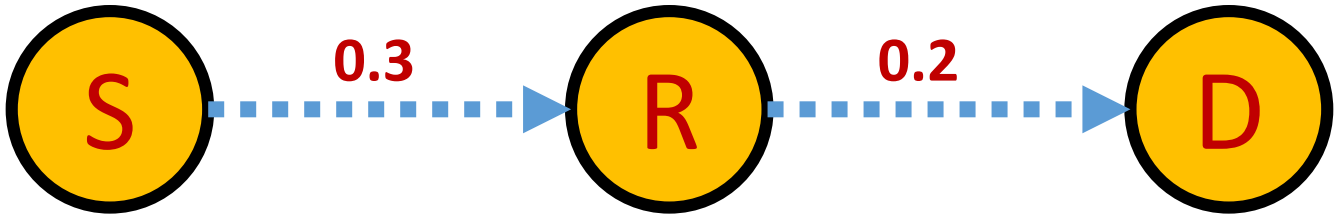
Task 2



Task 2

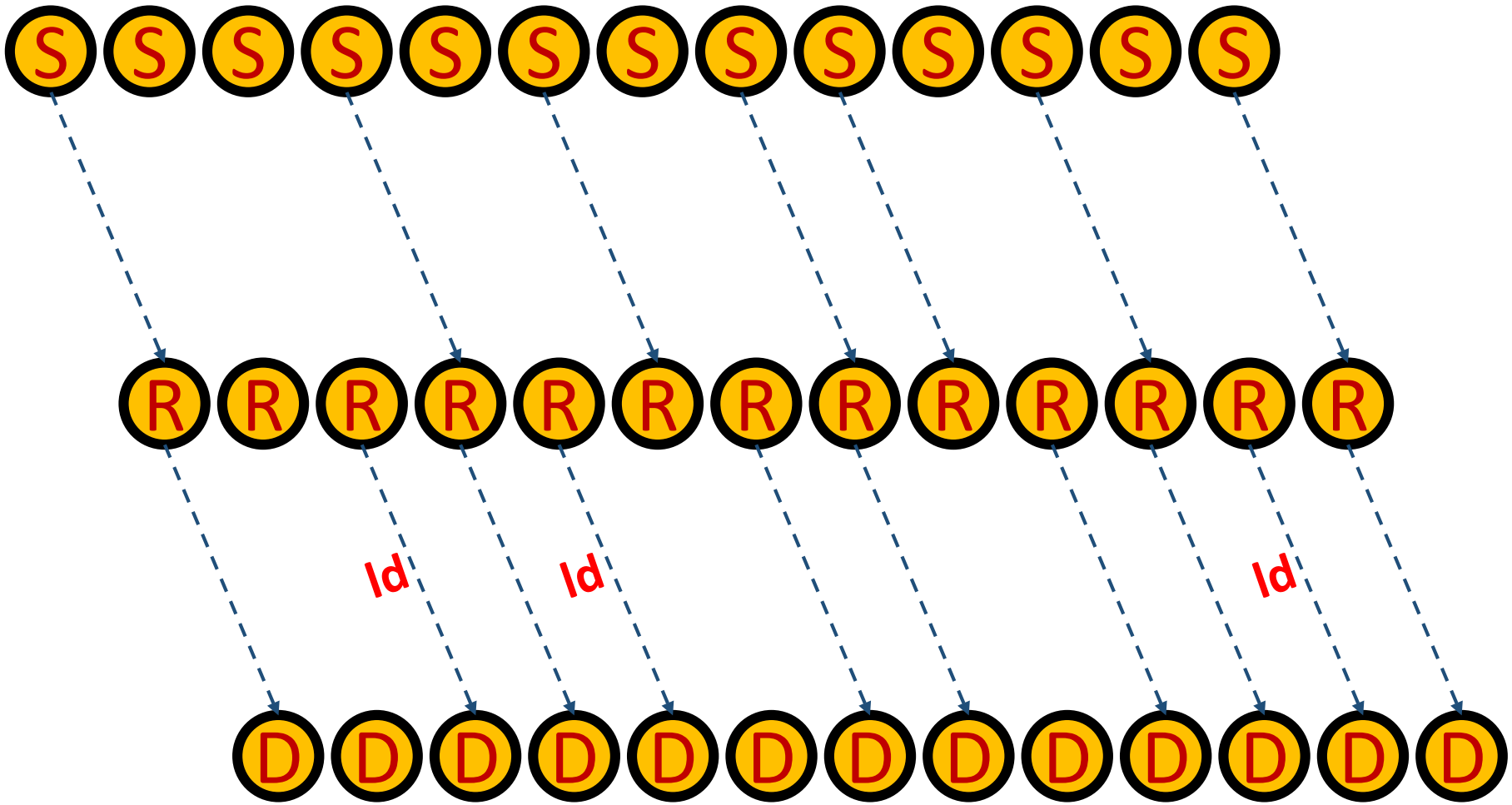


Task 2

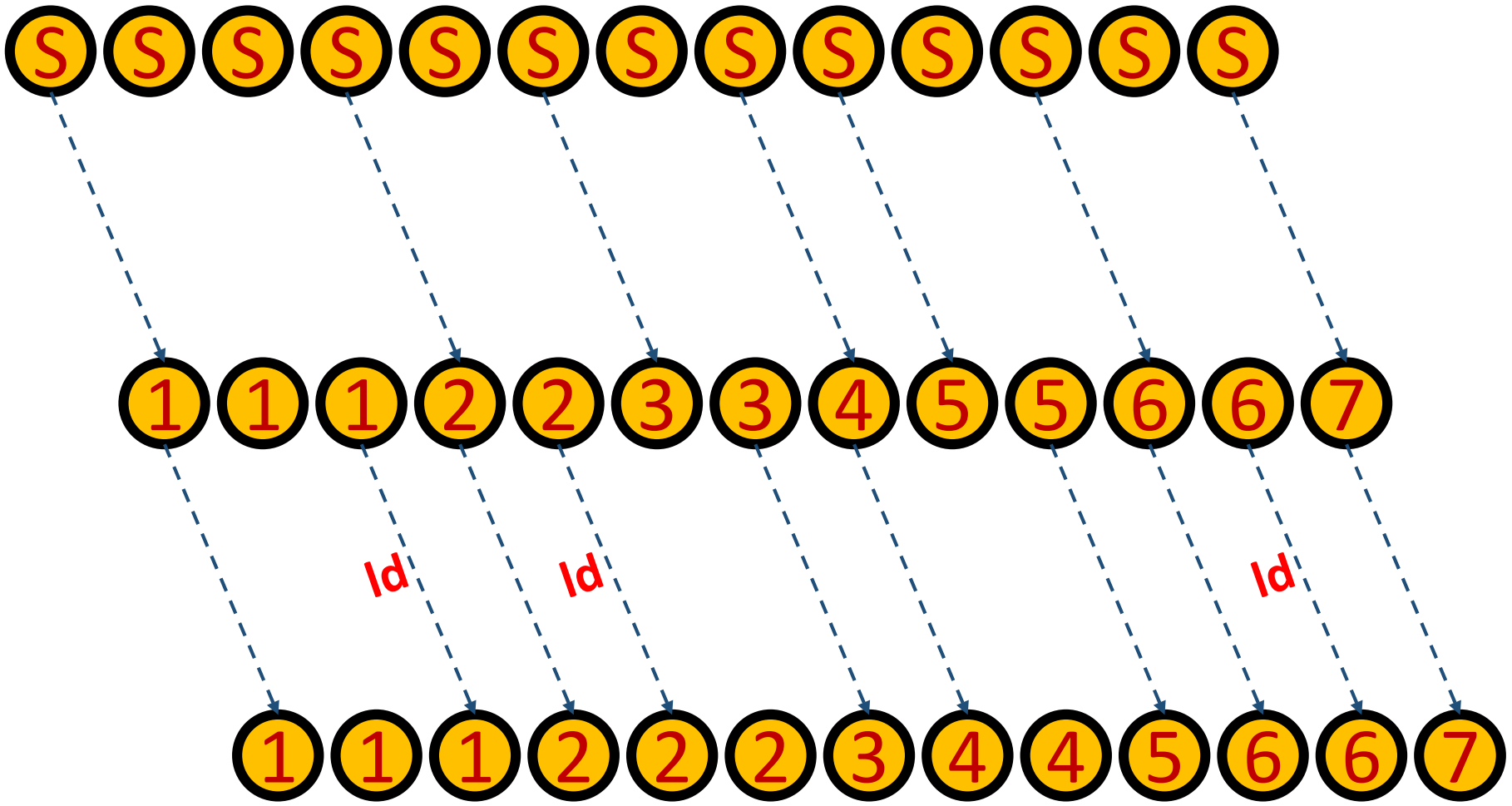


- Understanding that the relay has „nothing more to say“ as the incoming traffic is not increasing the rank fast enough.

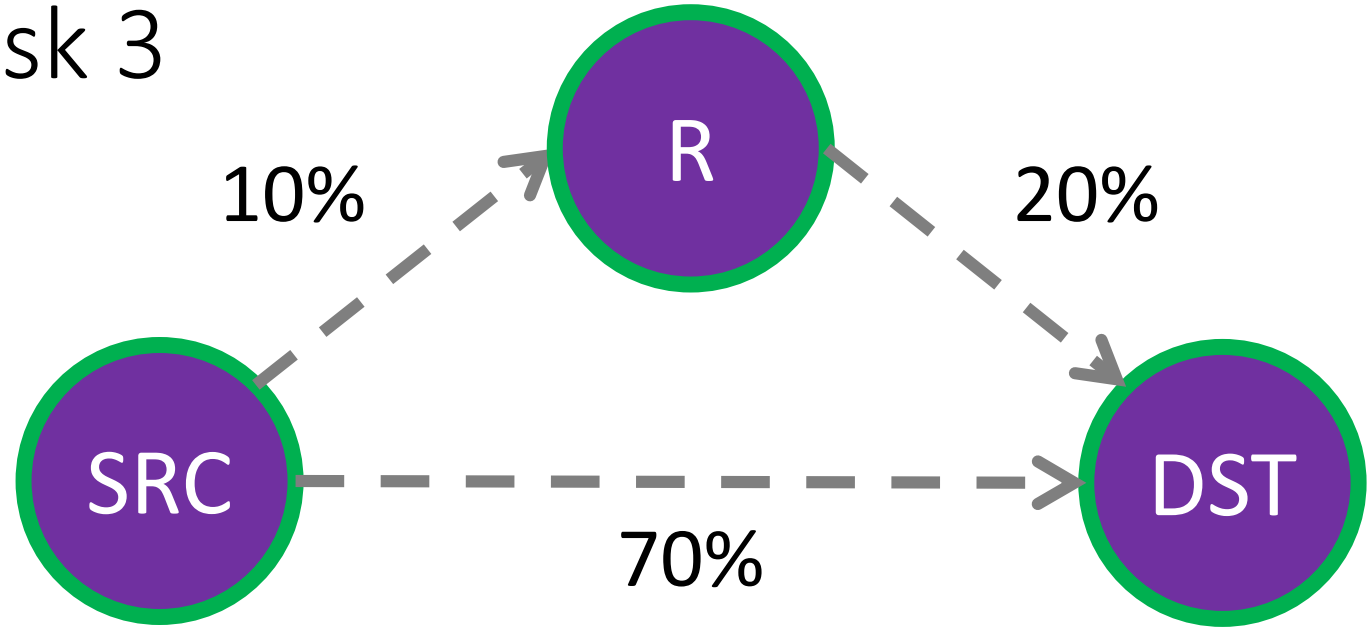
Task 2



Task 2



Task 3



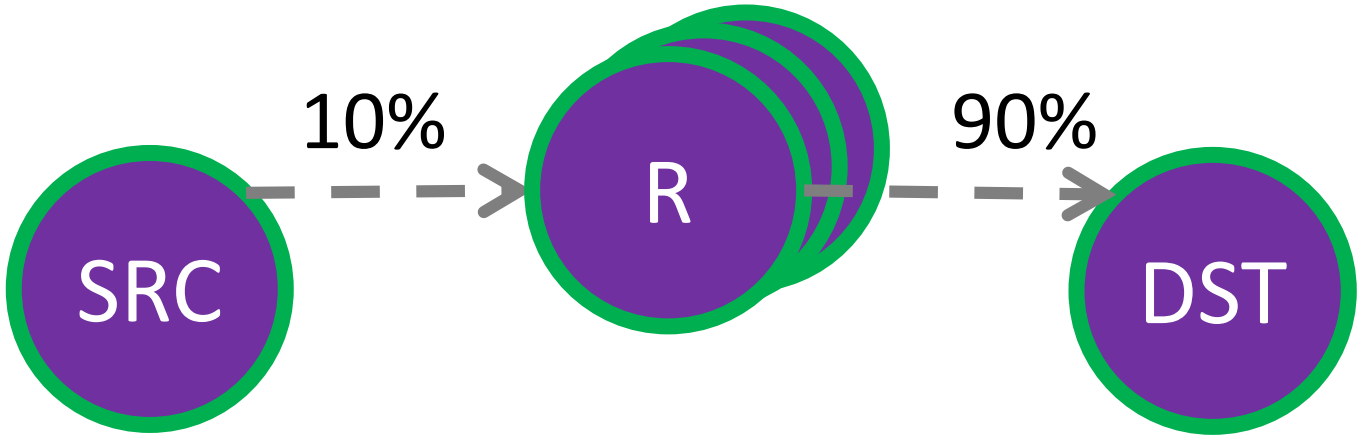
Parameter	Value
field	(binary|binary8|binary16)
source_systematic	0
symbols	8
symbols_size	100
runs	CHOOSE
recode	{0|1}
relays	1
error_source_sink	0.7
error_source_relays	0.1
error_relays_sink	0.2

Task 3

Parameter		Value
field	recode	Number of packets send by the source
binary	1	
binary 8	1	
binary 16	1	
binary	0	
binary 8	0	
binary 16	0	

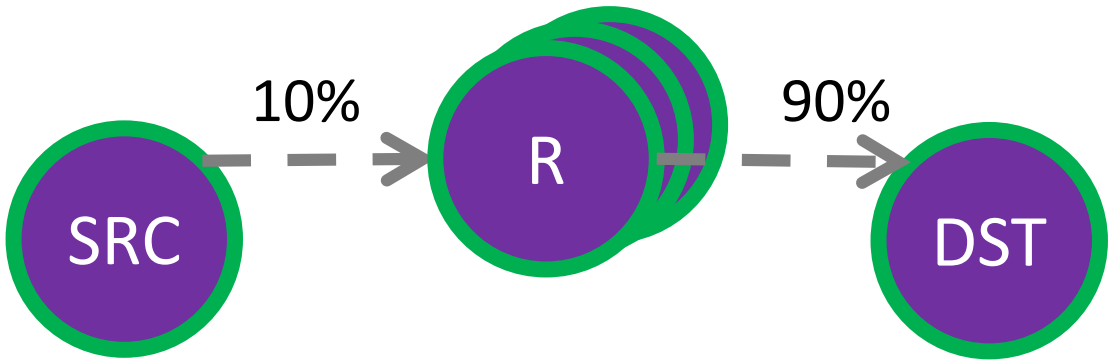
Parameter		Value
field	recode	Number of packets send by the source
binary	1	
binary 8	1	
binary 16	1	
binary	0	
binary8	0	
binary16	0	

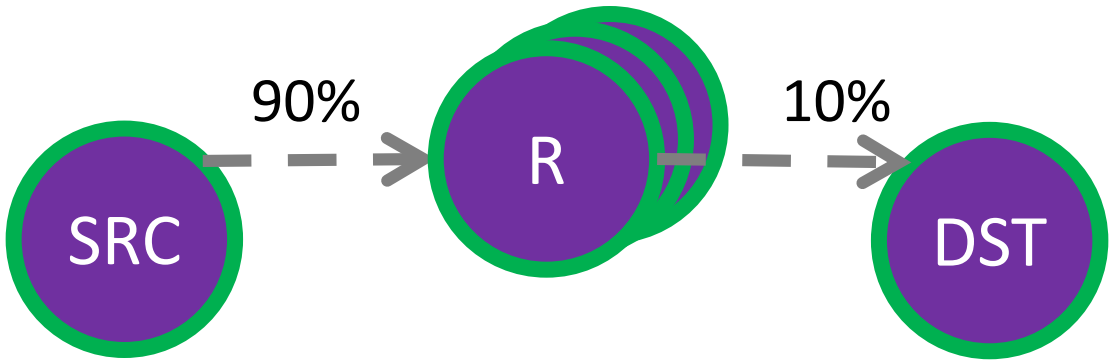
Task 4



Parameter	Value
field	(binary|binary8|binary16)
source_systematic	0
symbols	8
symbol_size	100
runs	CHOOSE
recode	CHOOSE
relays	CHOOSE
error_source_sink	1.0
error_source_relays	0.1
error_relays_sink	0.9

Task 4





Parameter		Value
R		Overall number of packets
1		
2		
3		
4		
5		
6		
7		

Parameter		Value
R		Overall number of packets
1		
2		
3		
4		
5		
6		
7		