University POLITEHNICA of Bucharest

Faculty of Automatic Control and Computers

Computer Science and Engineering Department

Formal Analysis of iptables Configurations

Scientific Student Projects Session – May 2017

Author(s)

Călin Cruceru

calin.cruceru@stud.acs.upb.ro

Scientific Advisor(s)

Conf.dr.ing Costin Raiciu

Motivation

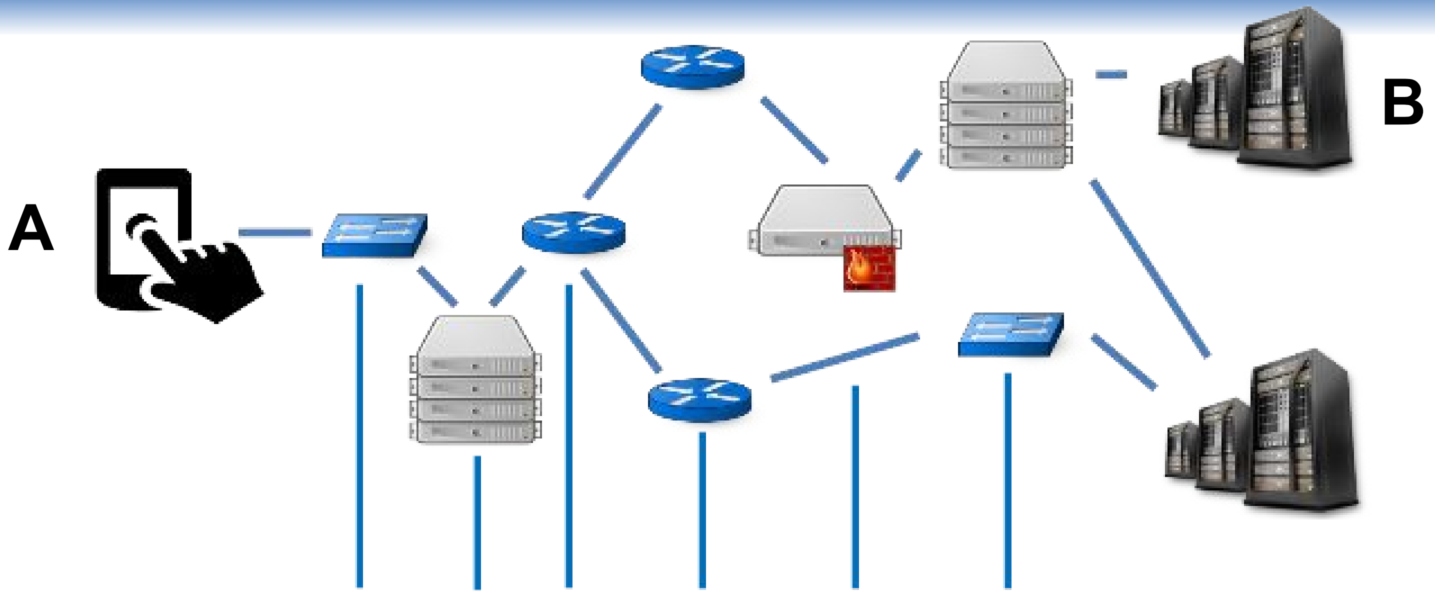
- Computer networks are increasingly complex
- Network Function Virtualization - a very promising trend
    - e.g. OpenStack's Neutron (**iptables**, Open vSwitch)

Motivation

- Computer networks are increasingly complex
- Network Function Virtualization - a very promising trend
  - e.g. OpenStack's Neutron (**iptables**, Open vSwitch)

- Formal methods developed for **network verification**
- Objective: verify even **more** networks

Static Verification - Overview



Data plane snapshot

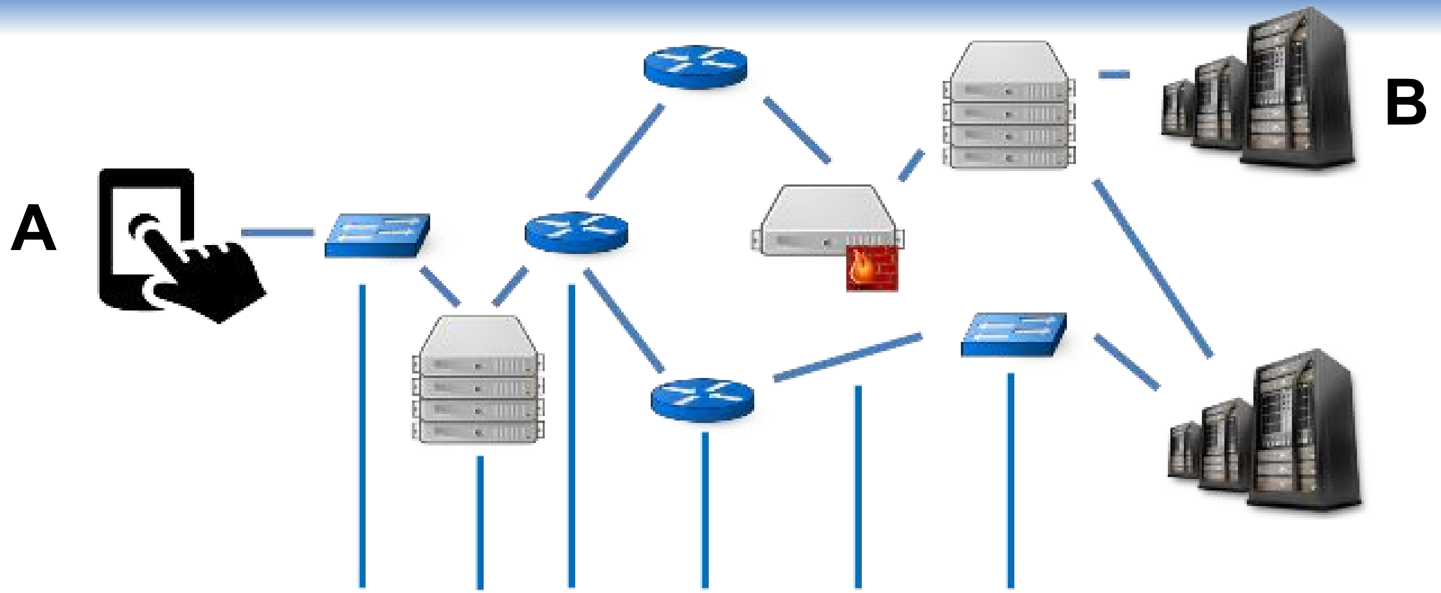
↓

Network model

↓

Verification engine

Static Verification - Overview



Data plane snapshot

↓

Network model - SEFL

↓

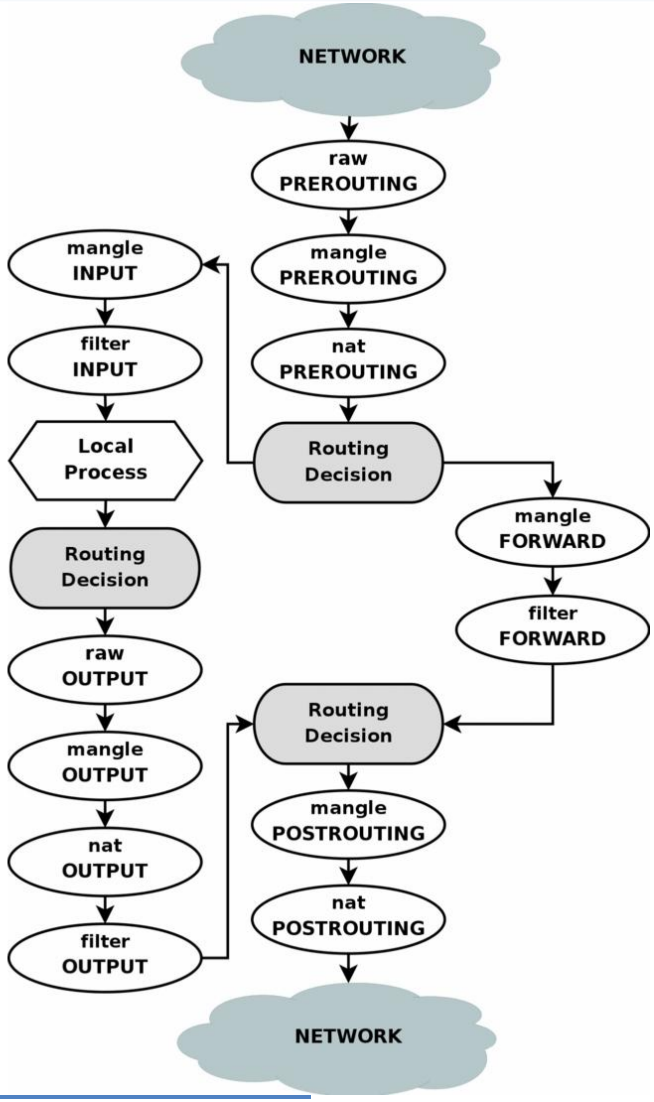
✓ ← Verification engine - SymNet → ✗

Static Verification - Framework

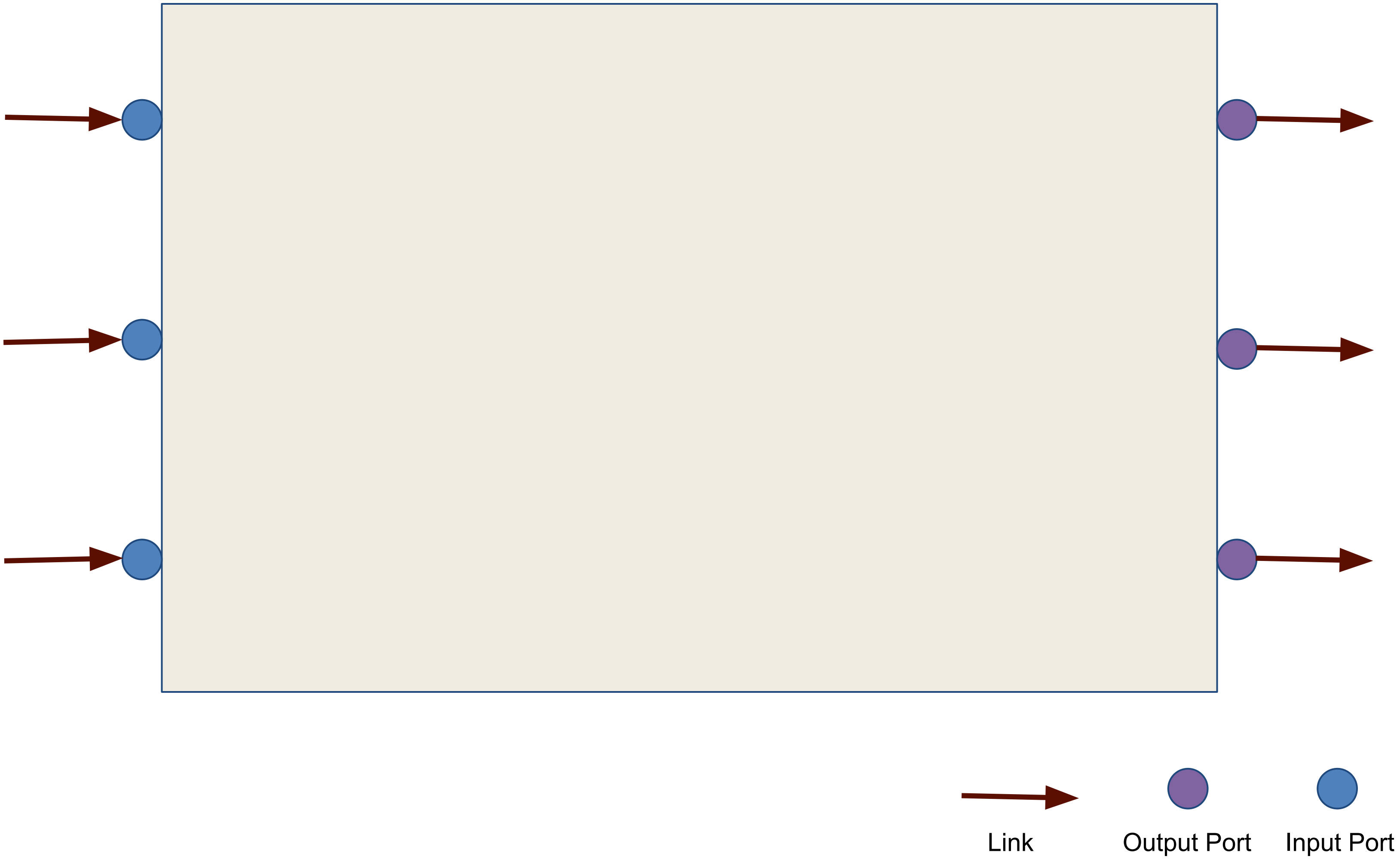
- SEFL (Symbolic Execution Friendly Language)
  - used to describe **network elements** as *flow transformations*
- SymNet (Symbolic network analysis tool)
  - input: network model
  - output: all possible symbolic paths
- Properties: **scalable**, memory safe
- Ready-made network models:
  - router/switch forwarding table, CISCO ASA firewall, Click modular router, OpenStack Neutron

iptables - Overview

- Tool used for packet filtering/mangling
- Based on Netfilter (Linux kernel framework)
- Organization:
  - rules
  - chains
    - built-in
    - user-defined
  - tables



iptables - Model

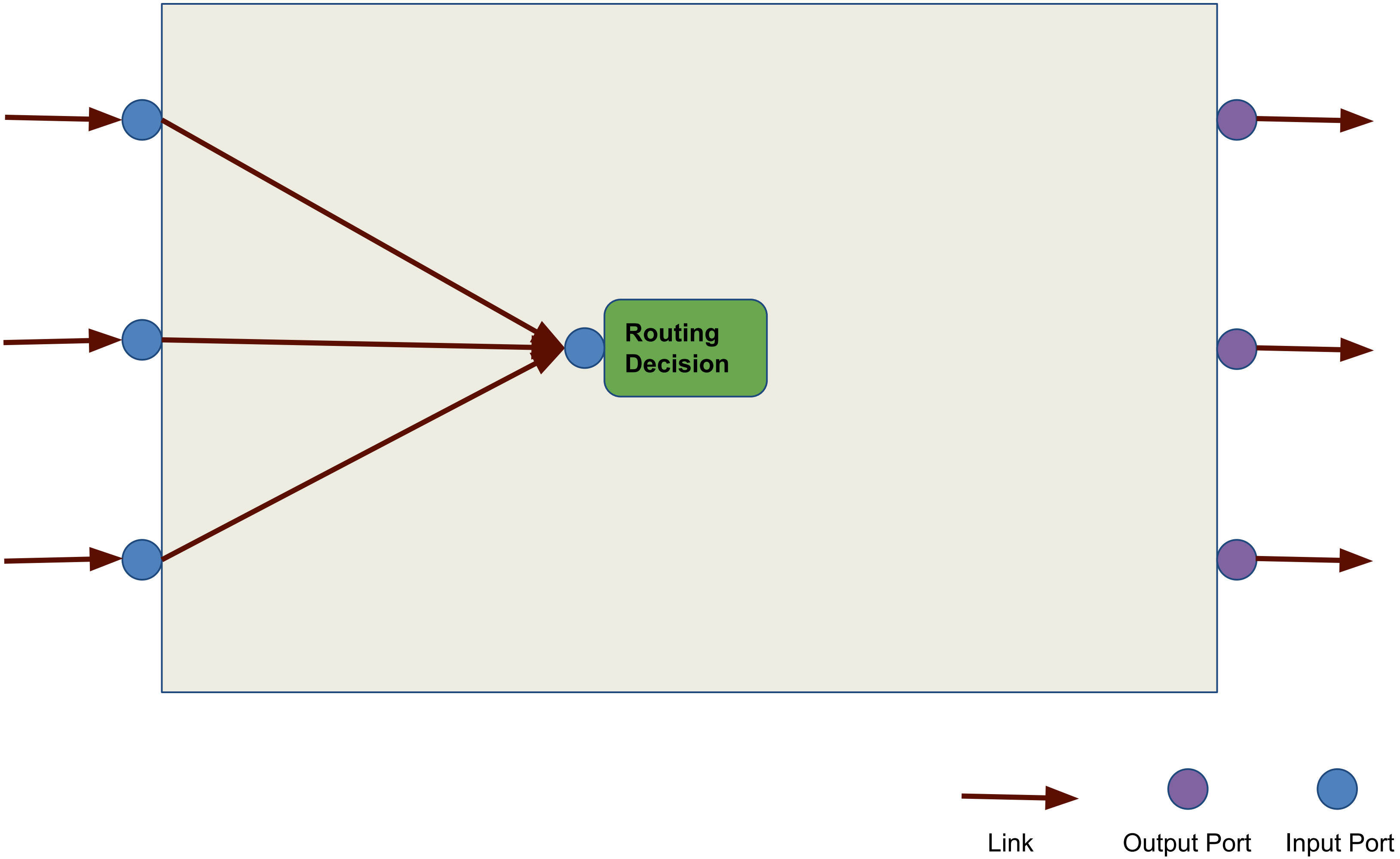


Link

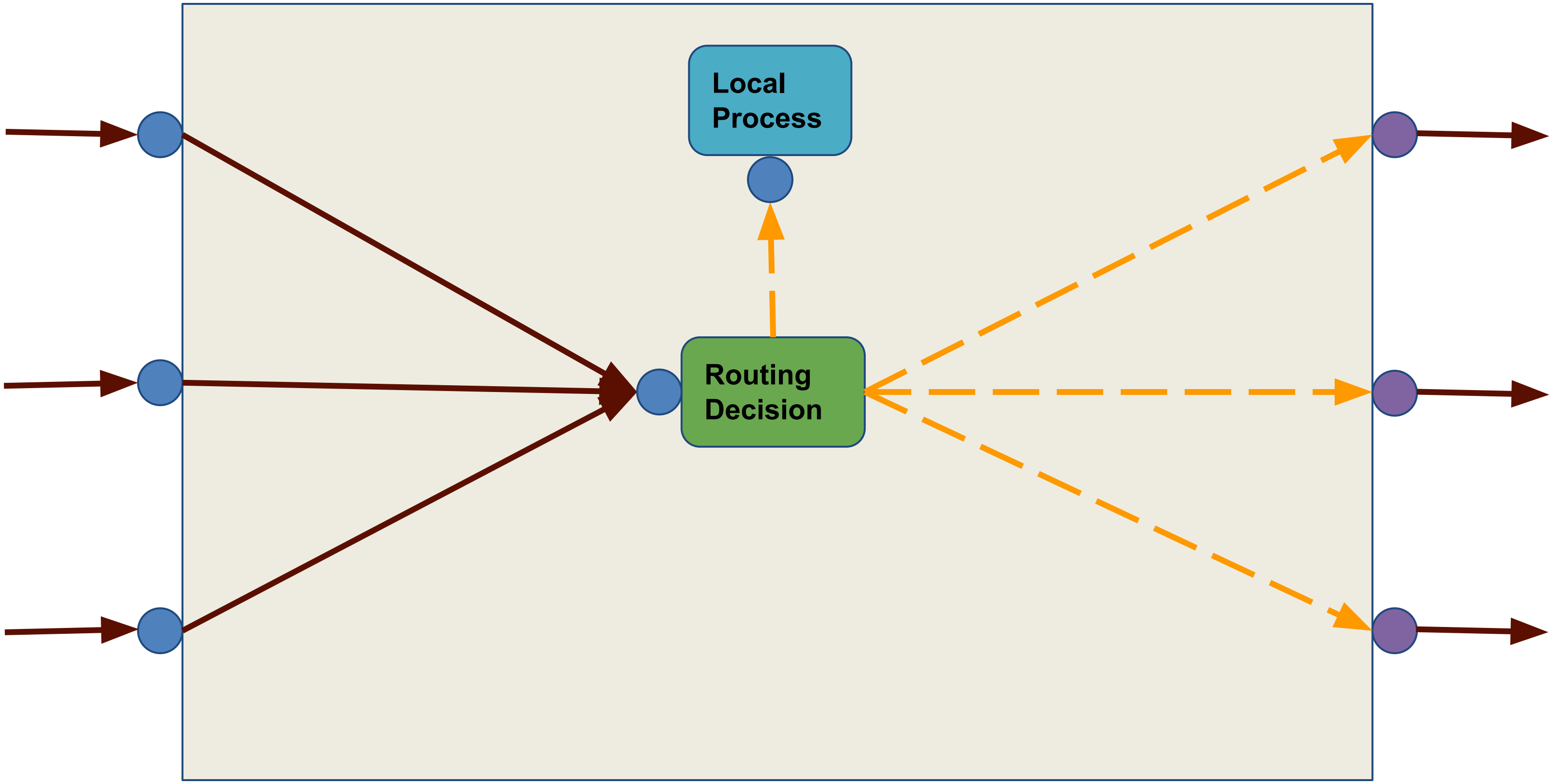
Output Port

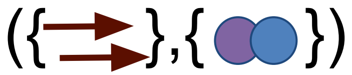
Input Port

iptables - Model

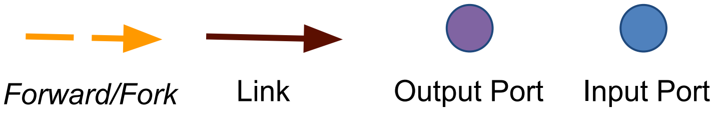


iptables - Model

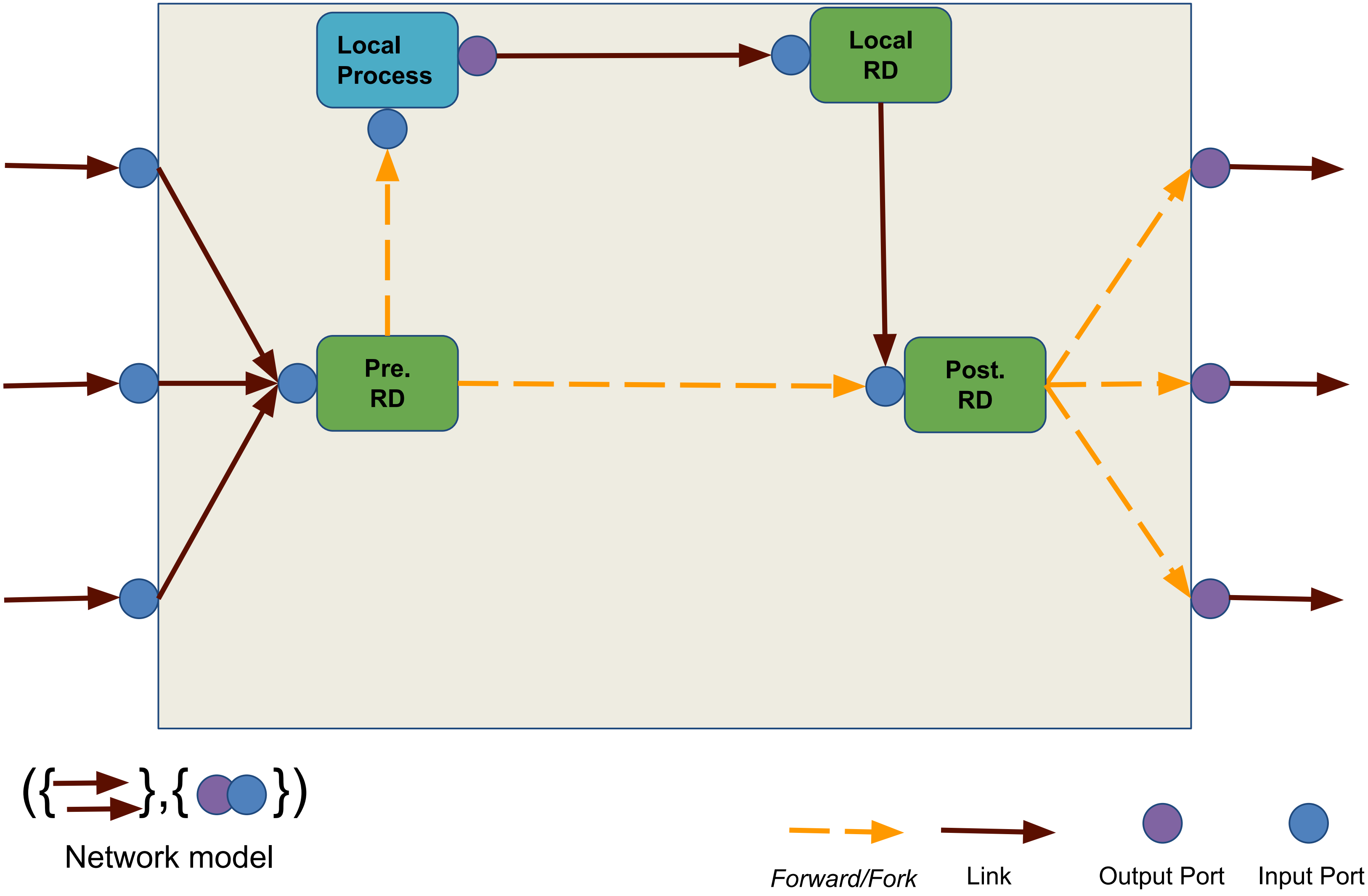




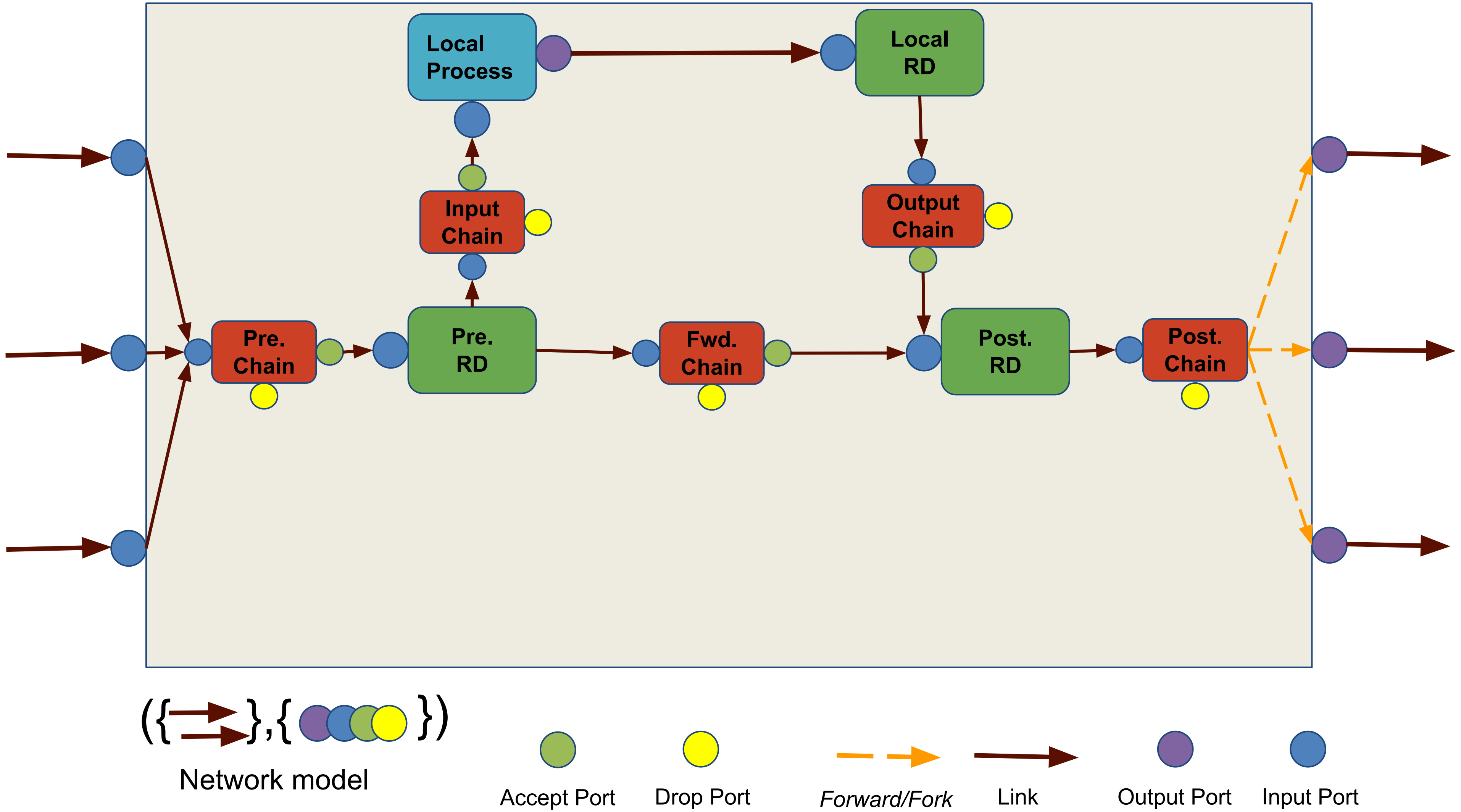
Network model



iptables - Model



iptables - Model



Local Process

Local RD

Input Chain

Output Chain

Pre. Chain

Pre. RD

Fwd. Chain

Post. RD

Post. Chain

Network model

Accept Port

Drop Port

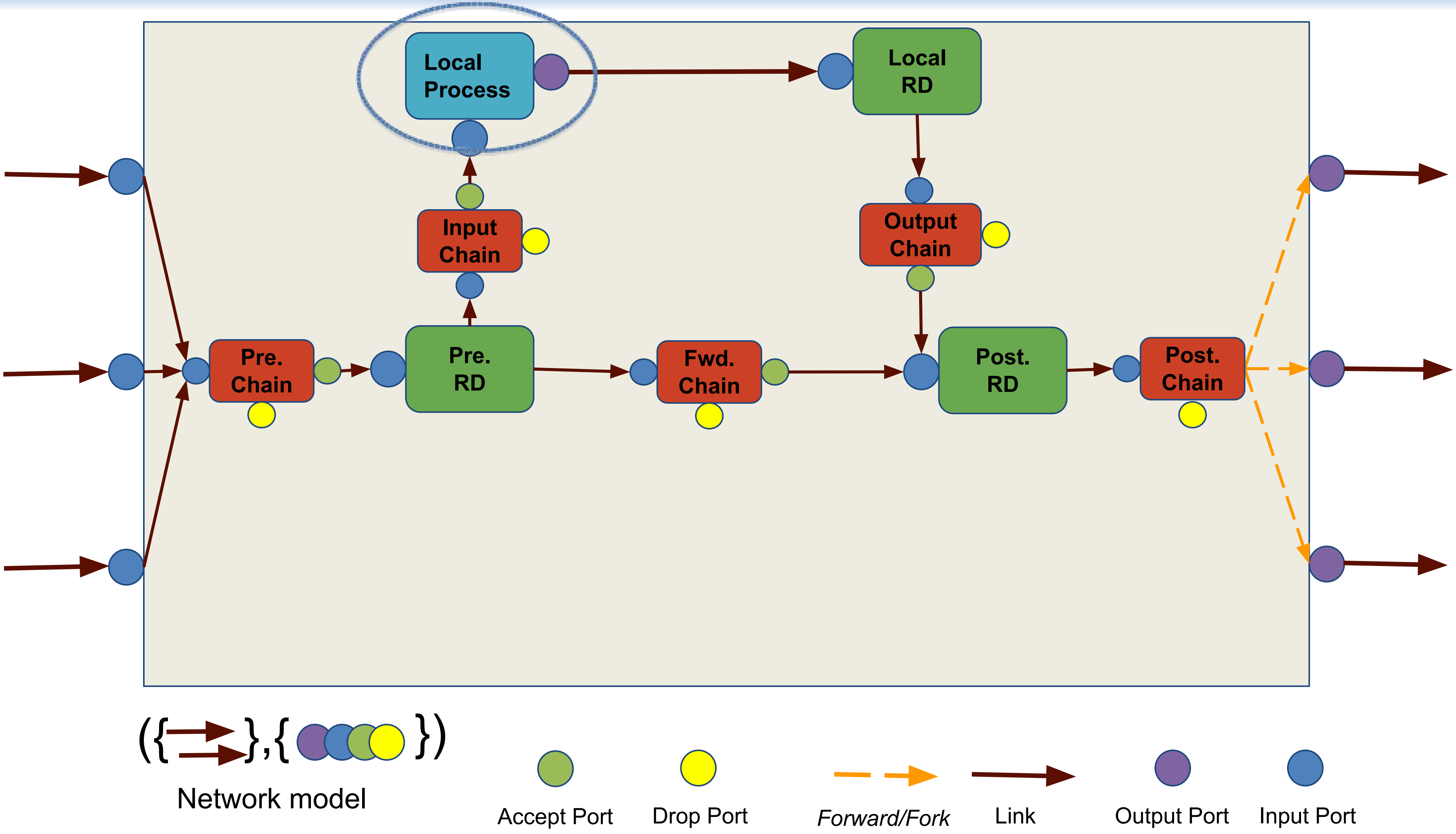
Forward/Fork

Link

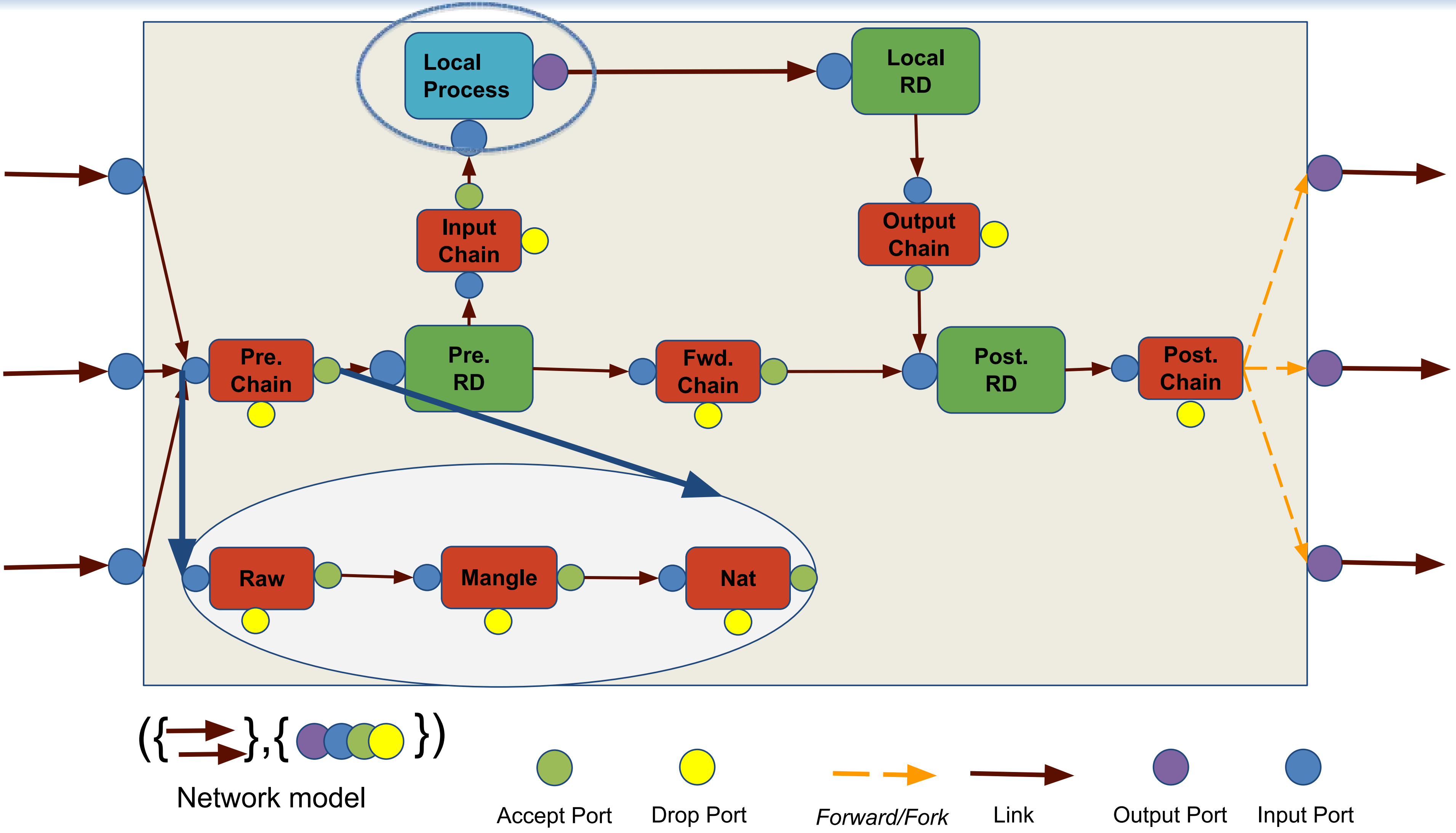
Output Port

Input Port

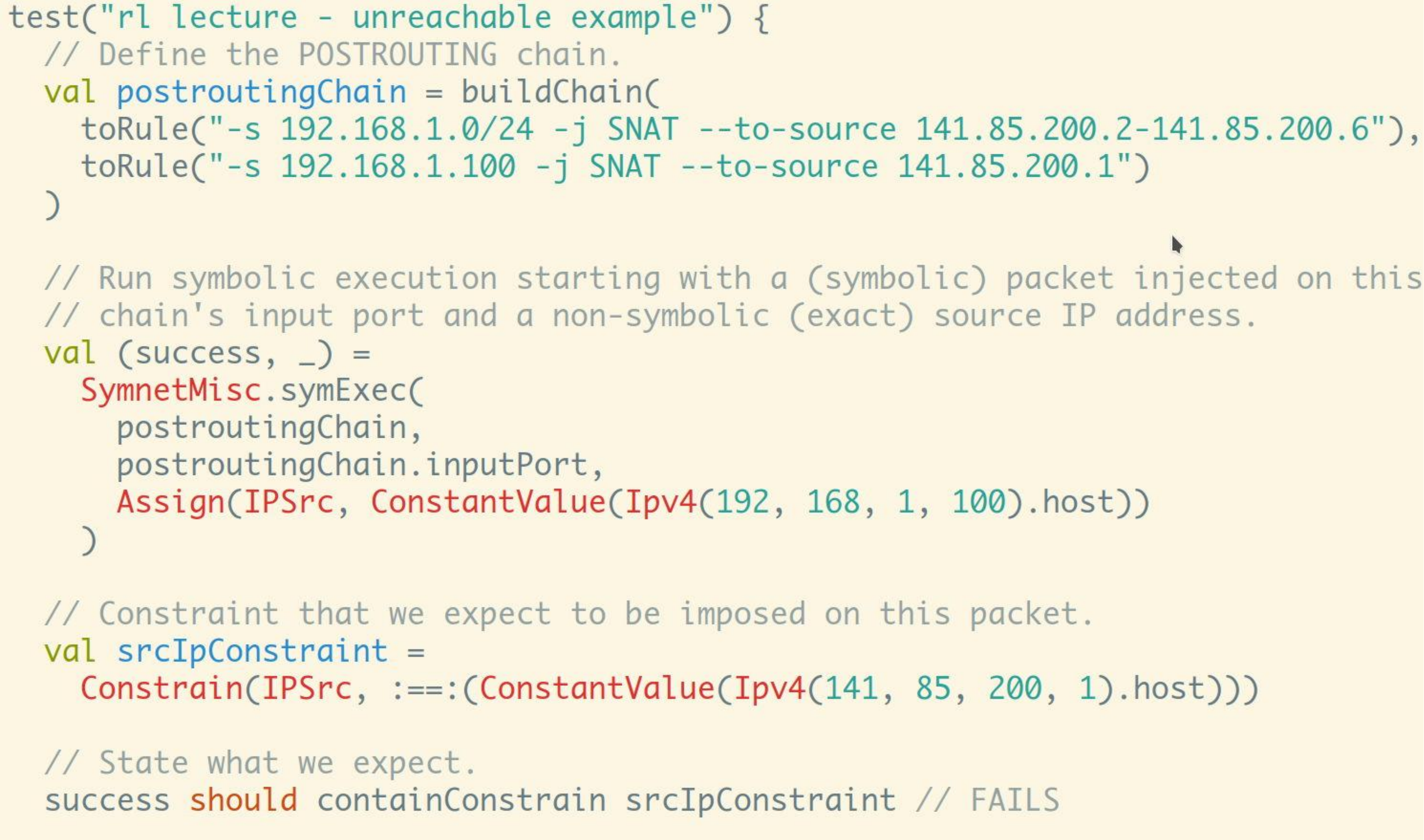
iptables - Model



iptables - Model



iptables - Example



```
test("rl lecture - unreachable example") {
  // Define the POSTROUTING chain.
  val postroutingChain = buildChain(
    toRule("-s 192.168.1.0/24 -j SNAT --to-source 141.85.200.2-141.85.200.6"),
    toRule("-s 192.168.1.100 -j SNAT --to-source 141.85.200.1")
  )

  // Run symbolic execution starting with a (symbolic) packet injected on this
  // chain's input port and a non-symbolic (exact) source IP address.
  val (success, _) =
    SymnetMisc.symExec(
      postroutingChain,
      postroutingChain.inputPort,
      Assign(IPSrc, ConstantValue(Ipv4(192, 168, 1, 100).host))
    )

  // Constraint that we expect to be imposed on this packet.
  val srcIpConstraint =
    Constrain(IPSrc, :==:(ConstantValue(Ipv4(141, 85, 200, 1).host)))

  // State what we expect.
  success should containConstrain srcIpConstraint // FAILS
```

Implementation & Future Work

- Compiler-like design
  - parsing, validation, (SEFL) code generation
  - easy to augment with new extensions
- Features implemented:
  - filter & NAT (SNAT/DNAT)
  - support for user-defined chains
  - apx. 4k Scala LOC
- Future work:
  - optimize SEFL code for chain traversal
  - connection tracking
  - further testing

References

- Stoenescu, Radu, et al. "SymNet: scalable symbolic execution for modern networks." Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference. ACM, 2016.
- Stoenescu, Radu, et al. "Symnet: Static checking for stateful networks." Proceedings of the 2013 workshop on Hot topics in middleboxes and network function virtualization. ACM, 2013.
- Stoenescu, Radu, et al. "In-Net: in-network processing for the masses." Proceedings of the Tenth European Conference on Computer Systems. ACM, 2015.
- GitHub repository: https://github.com/calincru/iptables-sefl
- http://www.iptables.info/
- netfilter Linux Kernel implementation - http://elixir.free-electrons.com/linux/latest/source/net/ipv4/netfilter