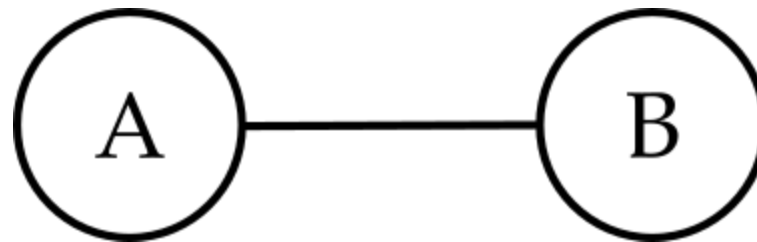


Simulating networks in your laptop

Maria Matejka • Oct 23rd, 2024

Two machines, one link (the most basic setup)

```
$ ./flock.py start test  
$ ./flock.py machine test A  
$ ./flock.py machine test B  
$ ./flock.py link test testlink A B
```



Login to a machine

```
$ ./flock.py shell test A  
Trying port 64122  
Trying ::1...  
Connected to localhost.  
Escape character is '^]'.
```

```
Linux 6.1.0-22-amd64 (A.000.test) (pts/12)
```

```
root@A:/run/user/1000/flock/hypervisors/test/000/A#
```

Networks

```
root@A:~# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 [...]
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: loc: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 [...]
    link/ether 8e:9c:c6:c7:56:84 brd ff:ff:ff:ff:ff:ff
3: testlink@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 [...]
    link/ether 5e:8b:97:f4:78:31 brd ff:ff:[...] link-netnsid 1
root@A:~#
```

Link autoseup

```
root@A:~# ip a show dev testlink
3: testlink@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 [...]
    link/ether 5e:8b:97:f4:78:31 brd [...]
    inet6 fe80::5c8b:97ff:fef4:7831/64 scope link
        valid_lft forever preferred_lft forever

root@B:~# ip a show dev testlink
3: testlink@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 [...]
    link/ether ce:8e:3b:17:4f:e9 brd [...]
    inet6 fe80::cc8e:3bff:fe17:4fe9/64 scope link
        valid_lft forever preferred_lft forever
```

Ping on link-locals

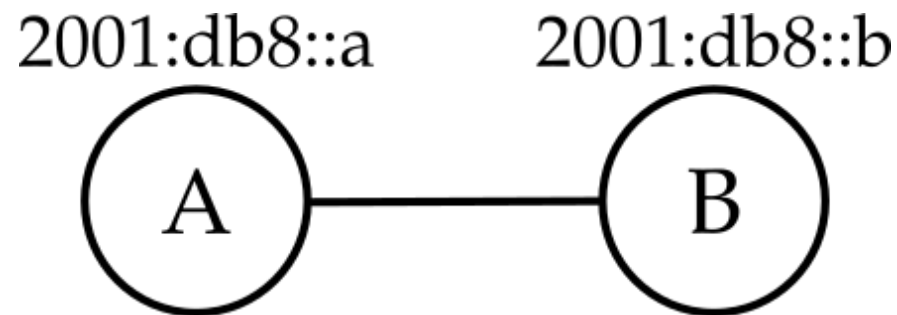
```
root@B:~# ping -c4 fe80::5c8b:97ff:fef4:7831%testlink
PING fe80::5c8b:97ff:fef4:7831%testlink([...]) 56 data bytes
64 bytes from fe80::5c8b:97ff:fef4:7831%testlink: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from fe80::5c8b:97ff:fef4:7831%testlink: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from fe80::5c8b:97ff:fef4:7831%testlink: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from fe80::5c8b:97ff:fef4:7831%testlink: icmp_seq=4 ttl=64 time=0.057 ms

--- fe80::5c8b:97ff:fef4:7831%testlink ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.020/0.052/0.076/0.020 ms
```

Global addresses, static routing

```
root@A:~# ip a add 2001:db8::a/128 dev loc
root@A:~# ip r add 2001:db8::b/128 via fe80::cc8e:3bff:fe17:4fe9
dev testlink
```

```
root@B:~# ip a add 2001:db8::b/128 dev loc
root@B:~# ip r add 2001:db8::a/128 via fe80::5c8b:97ff:fef4:7831
dev testlink
```



Dynamic routing with BIRD

```
root@A:~# cat bird.conf
router id 1; include "../common.conf";
root@B:~# cat bird.conf
router id 2; include "../common.conf";

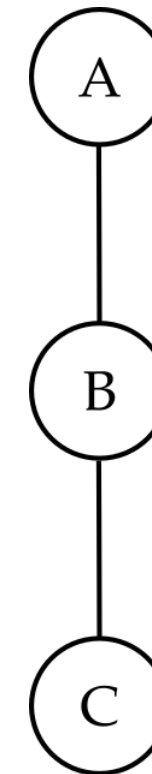
root@B:~# cat ../common.conf
protocol device {}
protocol kernel { ipv6 { export all; }; }
protocol ospf v3 { ipv6 { export all; };
    area 0 { interface "*link*";
        interface "loc" { stub; }; }; }
```

Adding more nodes

```
$ ./flock.py machine test C
$ ./flock.py link test clink C B

root@C:~# cat bird.conf
router id 3; include "../common.conf";

root@C:~# bird -l
root@C:~# ip a add 2001:db8::c/128 dev loc
```



Fixing common problems

- forwarding may be disabled by default

```
root@B:~# sysctl -w net.ipv6.conf.all.forwarding=1  
root@B:~# sysctl -w net.ipv4.conf.all.forwarding=1
```

- some packaged tcpdump builds don't work
(tcpdump built from sources works for me)

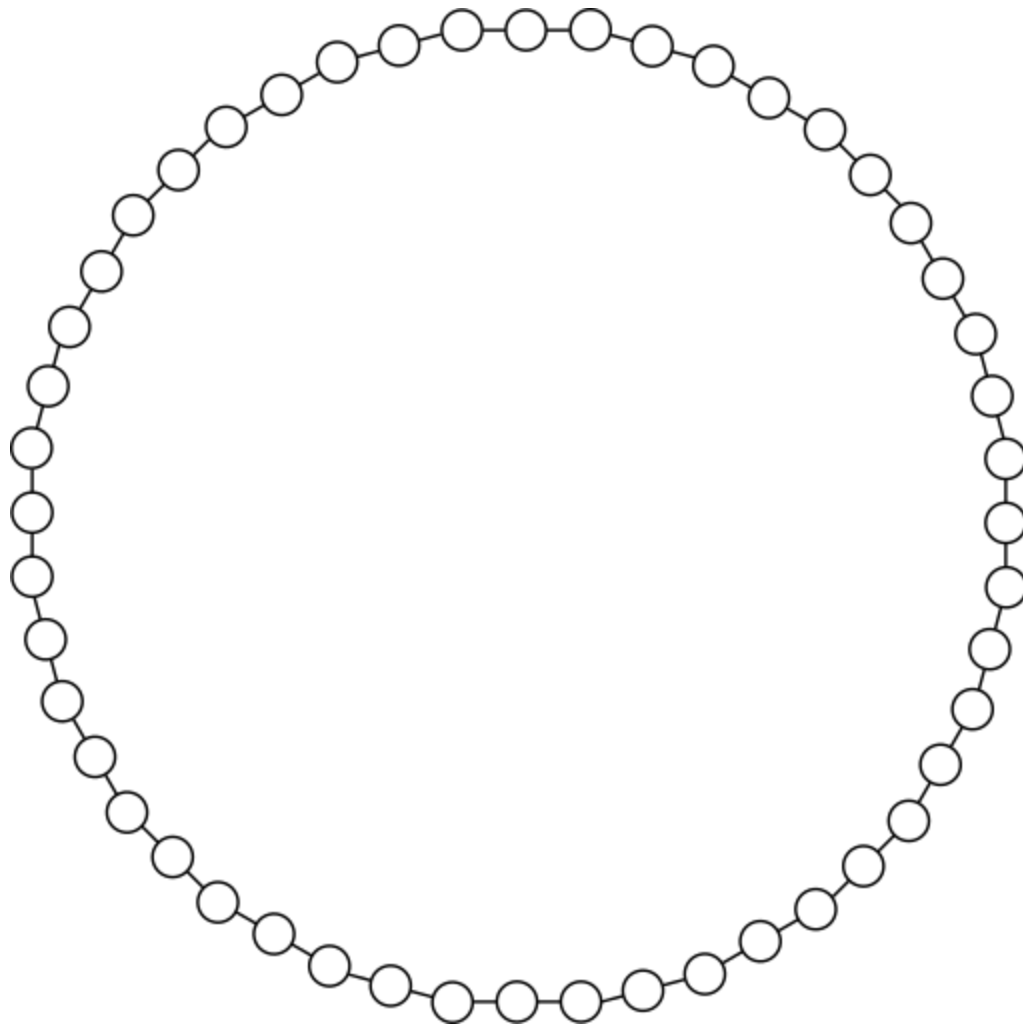
FRR config for B

```
frr version 8.4.4
frr defaults traditional
hostname B
log file frr.log
interface clink
    ipv6 ospf6 area 0
interface testlink
    ipv6 ospf6 area 0
router ospf6
    ospf6 router-id 1.2.3.4
    redistribute connected
```

Running FRR (locally built)

```
root@B:~# mkdir frr.socket
root@B:~# mkdir usr-local
root@B:~# mount --bind usr-local /usr/local
root@B:~# mkdir -p /usr/local/var/{lib,tmp}/frr
root@B:~# ~maria/frr/zebra/zebra -u root -g root -i zebra.pid --
vty_socket frr.socket -d -f frr.conf
root@B:~# ~maria/frr/ospf6d/ospf6d -u root -g root -i ospf6d.pid -
-vty_socket frr.socket -d -f frr.conf
```

Running 200 looped machines with OSPF



N=200

```
# Start all the machines
./flock.py start test
for m in $(seq $N); do
    ./flock.py machine test M$m
done
```

Running 200 looped machines with OSPF

```
# Prepare common configuration
cat <<EOF > /run/user/1000/flock/hypervisors/test/000/common.conf
protocol device {}
protocol kernel { ipv6 { export all; }; }
protocol ospf v3 { ipv6 { export all; };
    area 0 { interface "L*" { hello 2; };
        interface "loc" { stub; }; }; }
EOF
```

Running 200 looped machines with OSPF

```
# Prepare specific configurations
for m in $(seq $N); do
    cat <<EOF >
/run/user/1000/flock/hypervisors/test/000/M$m/bird.conf
router id $m;
include "../common.conf";
EOF
done
```

Running 200 looped machines with OSPF

```
# Run BIRD on all the machines
for m in $(seq $N); do
    (sleep 0.3
    cat <<EOF
sysctl -w net.ipv6.conf.all.forwarding=1
bird -l
ip a add 2001:db8::$(printf '%x' $m)/128 dev loc
EOF
    sleep 0.3) | ./flock.py shell test M$m
done
```

Running 200 looped machines with OSPF

```
# Link all the machines together
./flock.py link test L$N-1 M$N M1
for m in $(seq $((N-1))); do
    ./flock.py link test L$m-$((m+1)) M$m M$((m+1))
done
```

Running 200 looped machines with OSPF

```
# Show routes
root@M1:~# ip -6 r
2001:db8::62 via fe80::c25:2bff:fe92:58dc dev L1-2 proto bird [...]
2001:db8::63 via fe80::c25:2bff:fe92:58dc dev L1-2 proto bird [...]
2001:db8::64 via fe80::c25:2bff:fe92:58dc dev L1-2 proto bird [...]
2001:db8::65 proto bird [...]
    nexthop via fe80::c25:2bff:fe92:58dc dev L1-2 weight 1
    nexthop via fe80::9cc2:4ff:feb9:2c44 dev L200-1 weight 1
2001:db8::66 via fe80::9cc2:4ff:feb9:2c44 dev L200-1 proto bird [...]
2001:db8::67 via fe80::9cc2:4ff:feb9:2c44 dev L200-1 proto bird [...]
2001:db8::68 via fe80::9cc2:4ff:feb9:2c44 dev L200-1 proto bird [...]
```

Running 200 looped machines with OSPF

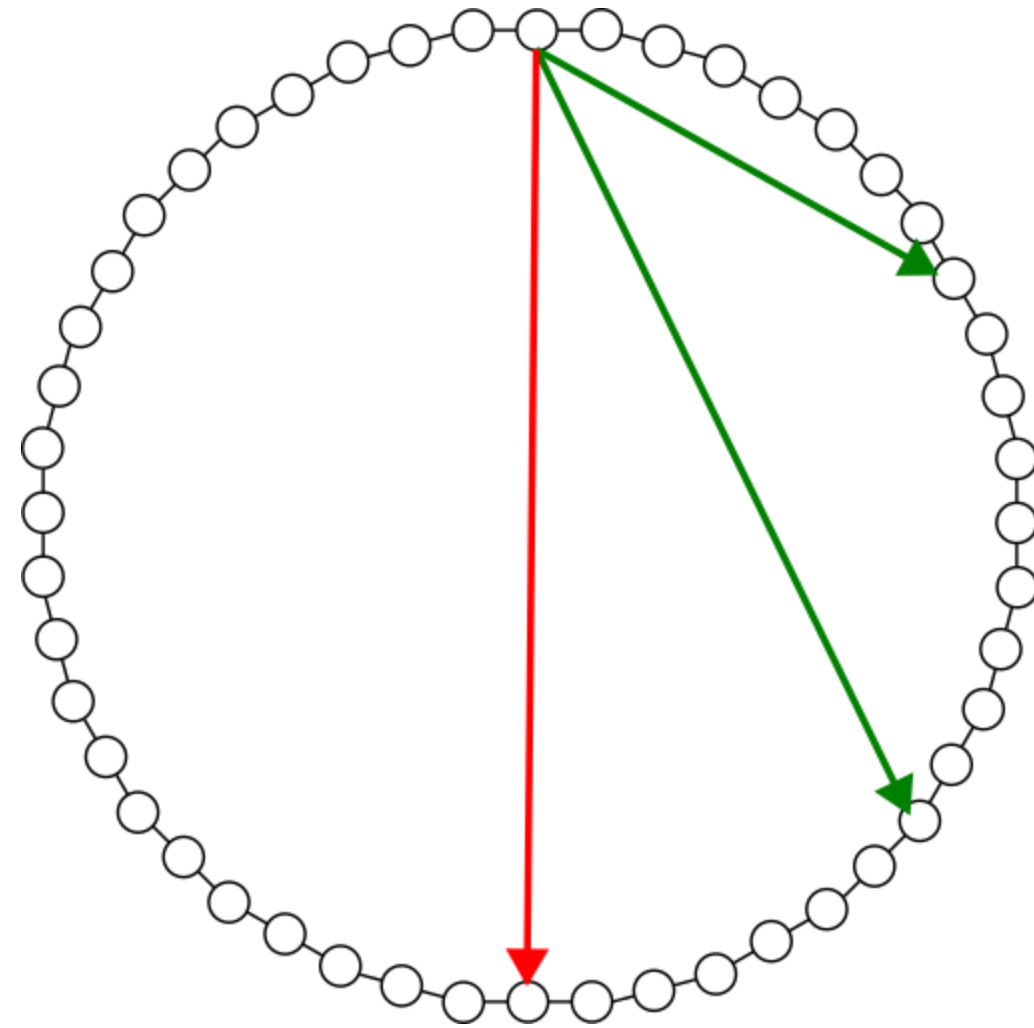
```
# Check reachability
root@M1:~# ping -c1 2001:db8::20
[...]
64 bytes from 2001:db8::20: icmp_seq=1 ttl=34 time=2.04 ms

root@M1:~# ping -c1 2001:db8::40
[...]
64 bytes from 2001:db8::40: icmp_seq=1 ttl=2 time=2.11 ms

root@M1:~# ping -c1 2001:db8::64
[...]
From 2001:db8::41 icmp_seq=1 Time exceeded: Hop limit
```

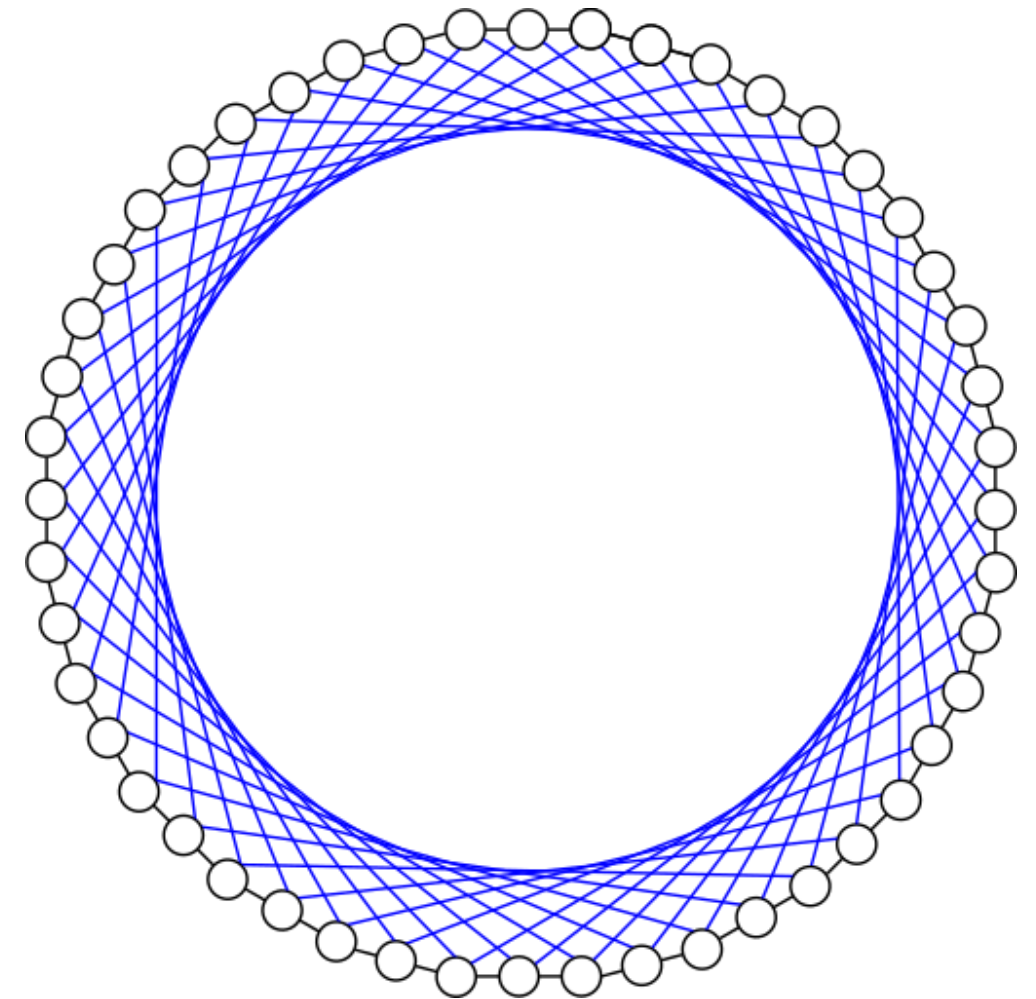
Running 200 looped machines with OSPF

Nodes are too far away, parts of the network are mutually unreachable



Running 200 looped machines with OSPF

```
K=30
# Loop is too big, add spines
for m in $(seq $(N-K)); do
    ./flock.py link test \
        L$m-$((m+K)) M$m M$((m+K))
done
for m in $(seq $K); do
    ./flock.py link test \
        L$((N-K+m))-$m M$((N-K+m)) M$m
done
```



Running 200 looped machines with OSPF

```
# Before: Too long paths
root@M1:~# ping -c1 2001:db8::64
[...]
From 2001:db8::41 icmp_seq=1 Time exceeded: Hop limit

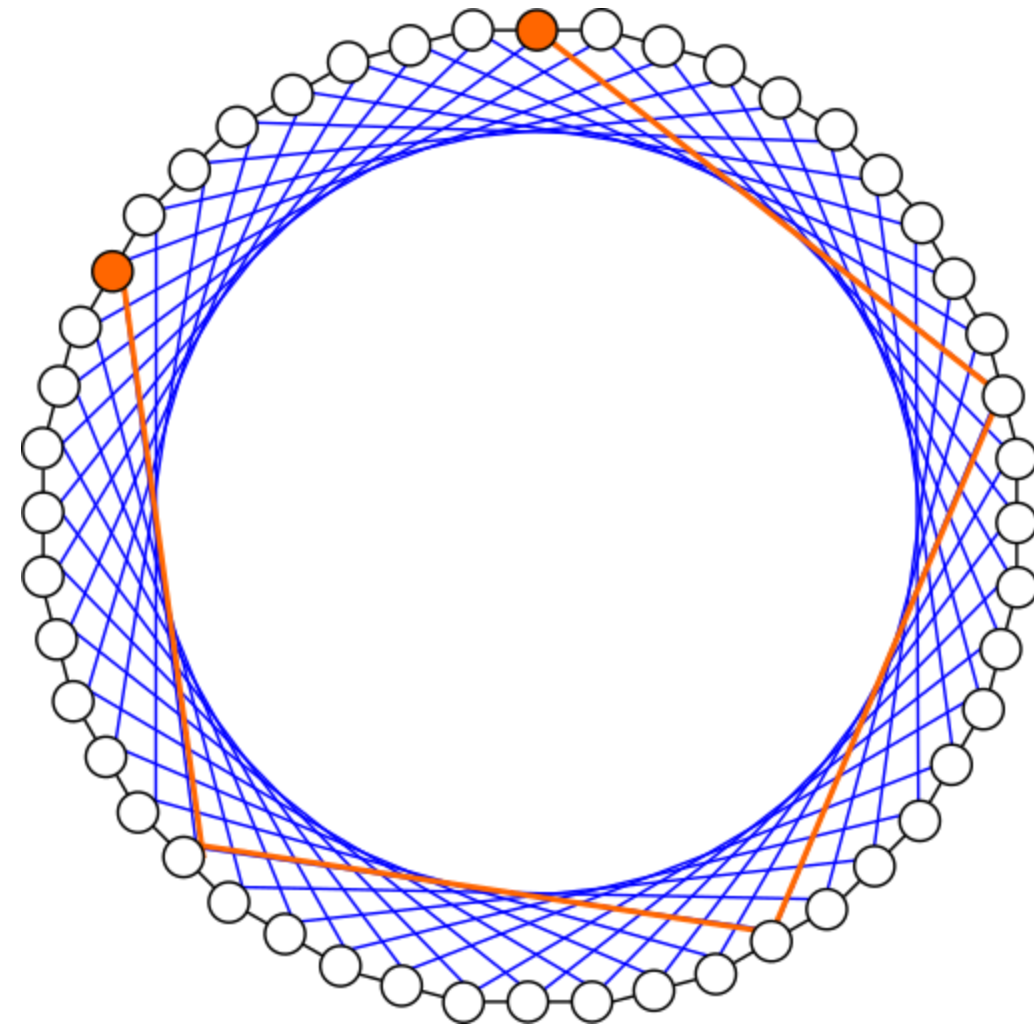
# After: Now it pings
root@M1:~# ping -c1 2001:db8::64
[...]
64 bytes from 2001:db8::64: icmp_seq=1 ttl=54 time=0.794 ms
```

Running 200 looped machines with OSPF

```
# We can show where it runs
root@M1:~# traceroute -n 2001:db8::64 -q 1
traceroute to 2001:db8::64 [...]
 1  2001:db8::1f    0.084 ms
 2  2001:db8::3d    0.063 ms
 3  2001:db8::3c    0.754 ms
 4  2001:db8::5a    0.124 ms
 5  2001:db8::78    0.135 ms
 6  2001:db8::96    0.145 ms
 7  2001:db8::b4    0.155 ms
 8  2001:db8::a     0.130 ms
 9  2001:db8::28    0.161 ms
10  2001:db8::46    0.154 ms
11  2001:db8::64    0.152 ms
```

Running 200 looped machines with OSPF

Shortest path now goes crazily around because of equal link costs.



Python package

```
import flock.Hypervisor as Hypervisor

h = Hypervisor.Hypervisor(definition="pyhv")
await h.prepare()
m = await asyncio.gather(*[
    h.machine(name, "minimalist")
    for name in ("A", "B", "C", "D", "E")
])

print(m[2].run_in("ip", "a"))
```

Flock: Conclusion

- fits well into a laptop
- 100's of easily accessible nodes
- repeatable and automatable
- purely non-root, deployable in university labs
- credits: Ondřej Zajíček for original Netlab

Flock: Future expectations

- support for running LXC/Docker images and QEMU (may need root though)
- clickable GUI (like GNC3 has)
- simulated packet loss, latency, ...
- interconnecting multiple instances
- connecting physical machines from outside

Flock: Tryout

- available at <https://gitlab.nic.cz/labs/flock>
- complete rework to C already in progress (yes, we are crazy)
- UI and API may change slightly
- please consult before contributing

Contacts

Maria Matejka

BIRD Team Leader at CZ.NIC

maria.matejka@nic.cz

@marenamat at social networks

BIRD Users Mailing list: bird-users@network.cz

BIRD Support: <https://bird.nic.cz/>

Questions, Explanations, Discussion

Maria Matejka • Oct 23rd, 2024