



RINA-enabled applications

NEXXTWORKS
ENGINEERING FORWARD

Vincenzo Maffione,
Marco Capitani (Presenter)
23/05/2018



Large scale RINA Experimentation on FIRE +

- A POSIX-like APIs for applications:
 - Reminiscent of the socket API, to ease porting of existing socket applications...
 - ... yet with the full power of RINA API (QoS support and complete naming scheme)
 - Easy to learn for network developers
 - Documentation available at
 - <https://github.com/rlite/rlite#9-rina-api-documentation>
 - <https://github.com/rlite/rlite/blob/master/include/rina/api.h>
- The API is implemented by both rlite and IRATI prototypes
 - Applications compile and run on both RINA implementations

- Main API calls:
 - `int rina_open()` → `fd`
 - Opens a control device instance, returning a file descriptor.
 - `int rina_flow_alloc(dif_name, local_name, remote_name, flowspec, flags)` → `fd`
 - Issues a flow allocation request and possibly wait for the associated response. Returns a file descriptor to be used for data transfer.
 - `int rina_register(fd, dif_name, appl_name, flags)`
 - Register an application into a given DIF.
 - `int rina_unregister(fd, dif_name, appl_name, flags)`
 - Unregister an application from a given DIF.
 - `int rina_flow_accept(fd, flags)` → `remote_appl, flowspec`
 - Wait and possibly accept an incoming flow request, where the destination application is one of the ones registered to the control device referred by `fd`. Returns a file descriptor to be used for data transfer.

Current RINA applications



- New applications (included in <https://github.com/r-lite/r-lite>)
 - **rinaperf**: multi-threaded client/server capable of parallel flow allocation, implementing basic connectivity testing, throughput and latency measurements
 - **rina-gw**: daemon acting as an application gateway between a RINA network and an IP network
 - **iporinad**: daemon that tunnels IP traffic over a RINA network, similarly to IP over MPLS
- Ported applications
 - **nginx**: RINA port of the popular Nginx web server
 - Code: <https://github.com/r-lite/rina-nginx>
 - **dropbear**: RINA port of the Dropbear ssh client and server
 - Code: <https://github.com/r-lite/rina-dropbear>

- Tool to measure performance of RINA networks
 - Role similar to **netperf** and **iperf** for TCP/IP networks
- Request response transaction rate used to measure latency

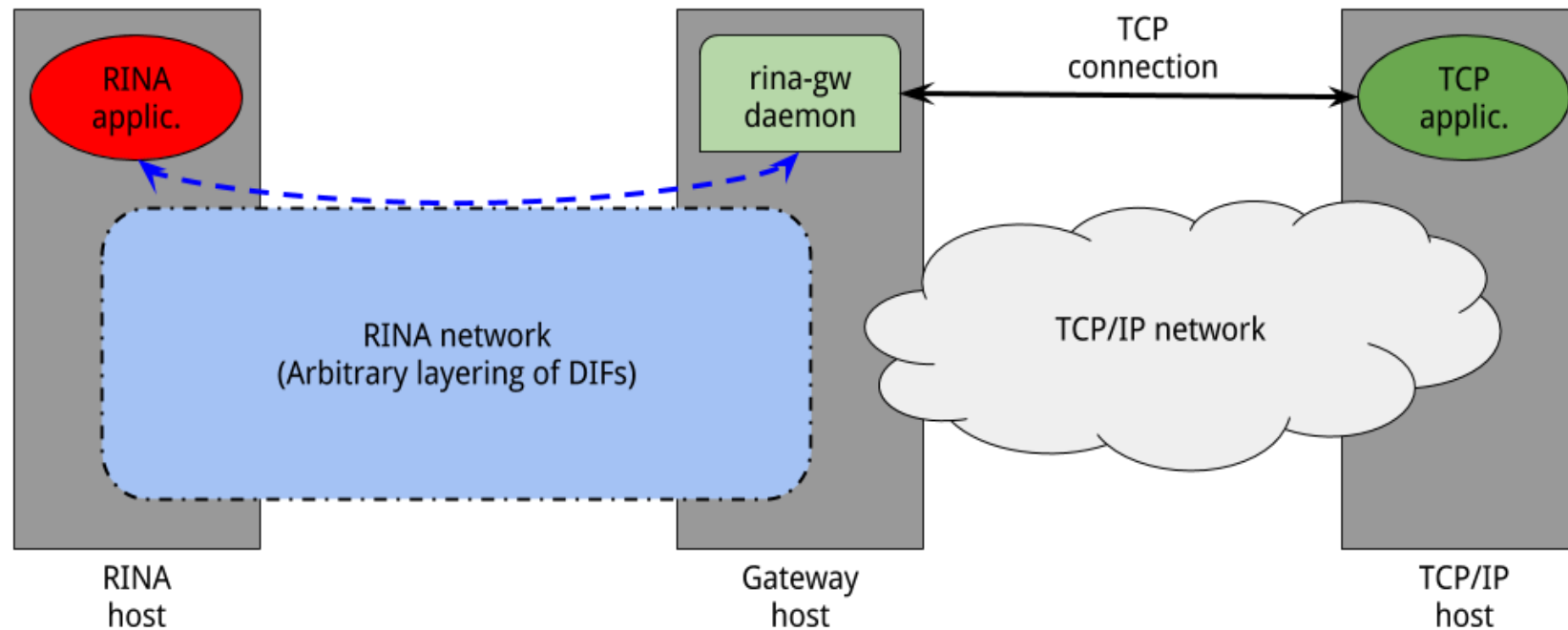
```
user@host ~/rina # rinaperf -c 1000000 -t rr -s 400
Starting request-response test; message size: 400, number of messages: 1000000, duration: inf
      Transactions      Kpps      Mbps      Latency (ns)
Sender 1000000          145.569    465.821    6869
```

- Unidirectional bandwidth test used to measure throughput and goodput

```
user@host ~/rina # rinaperf -t perf -s 1460 -D 5
Starting unidirectional throughput test; message size: 1460, number of messages: inf, duration: 5 secs
      Packets      Kpps      Mbps
Sender 6790377      1358.417    15866.311
Receiver 5037989      988.051     11540.436
```

rina-gw (1)

- A proxy between a TCP/IP network and a RINA network
 - Accept TCP connections coming from a TCP/IP network and proxy them by allocating RINA flows towards the proper server applications in the RINA network
 - Accepts flow allocation requests coming from the RINA network and proxies them to a TCP server using new TCP connections.

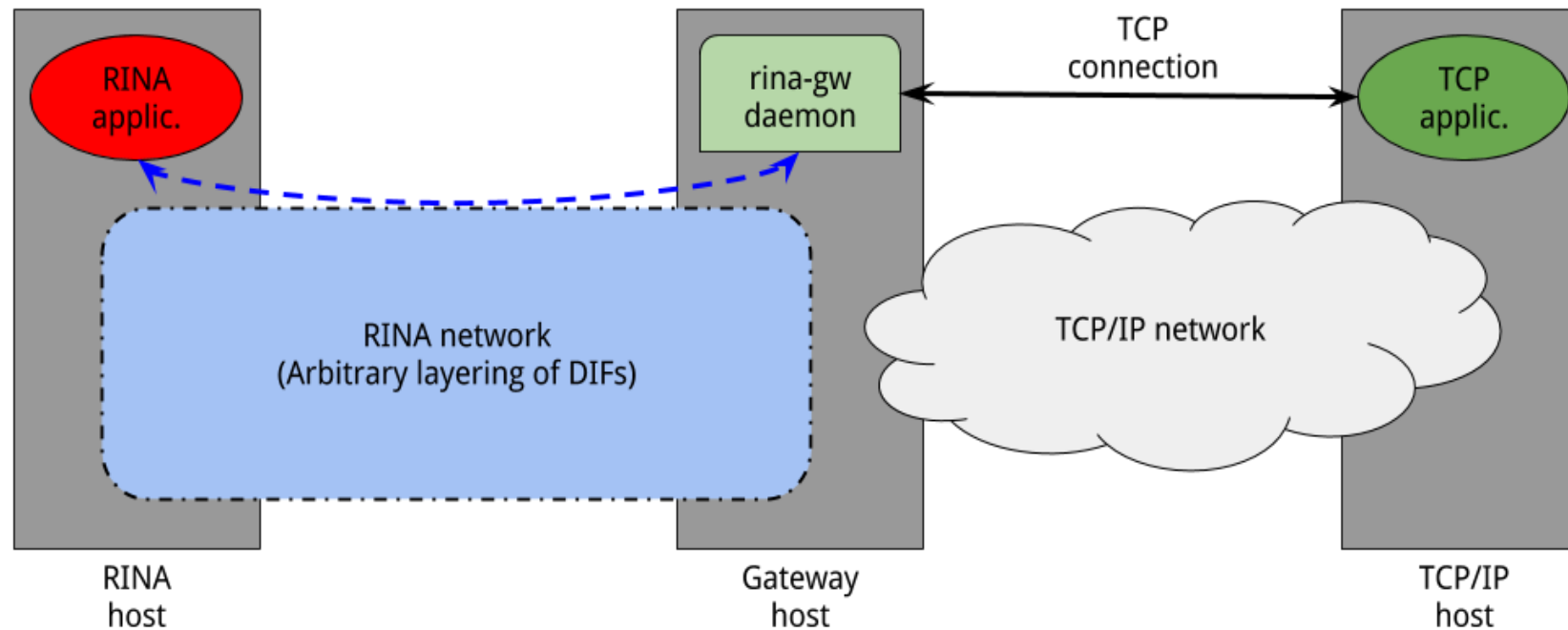


rina-gw (2)

- Mapping between TCP ports and RINA application names stored in a configuration file
 - Example:

```
I2R serv.DIF rina-nginx-1 0.0.0.0 80
```

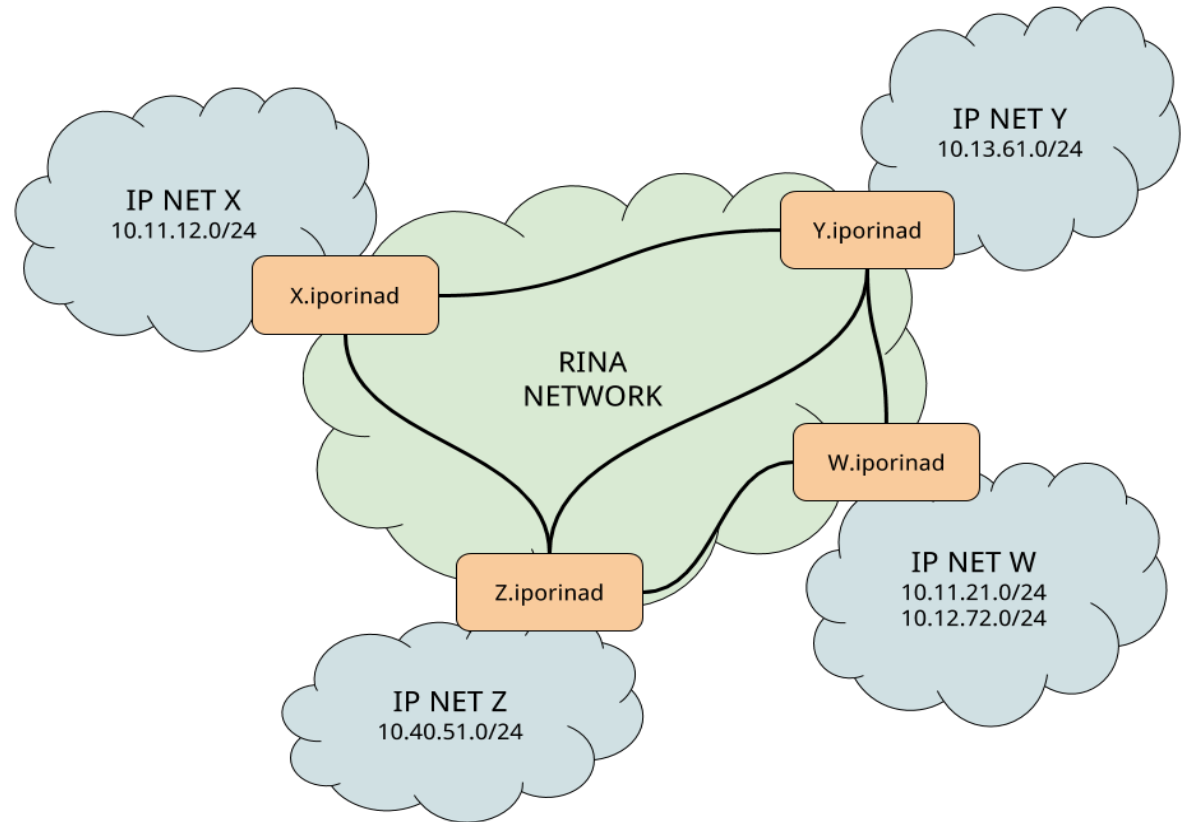
```
R2I vpn3.DIF gmail 32.1.42.190 8000
```



iporinad (1)

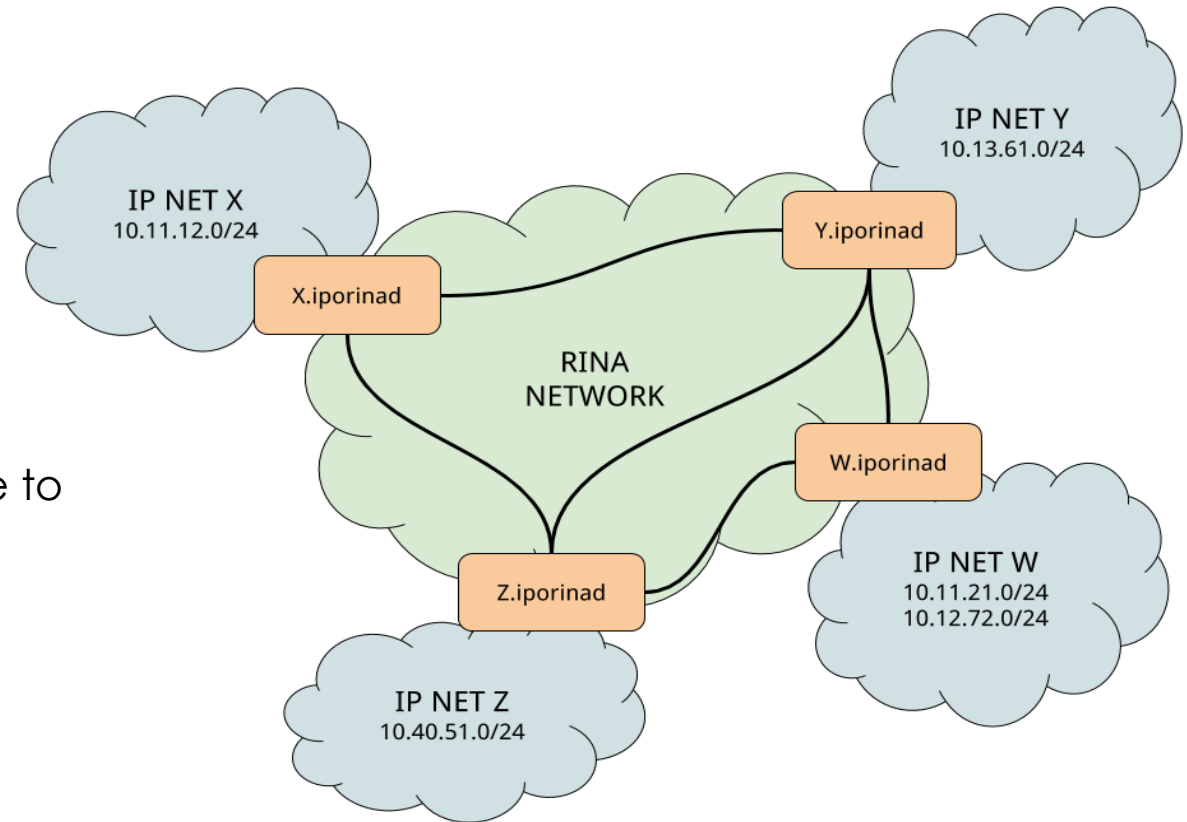


- An OpenVPN-like daemon that tunnels IP traffic over a RINA network
 - Role similar to MPLS within traditional IP/MPLS deployments
- Runs on the *edge* nodes at the boundary between an IP network and a RINA network
 - IP packets are encapsulated into or decapsulated from a RINA flow.



iporinad (2)

- Each iporinad instance advertises locally reachable IP subnets to its peers
- An instance is configured through a local configuration file, specifying
 - List of peers to connect to, together with a subnet for the IP tunnel (usually /30)
 - List of locally reachable IP subnet to advertise to all the peers

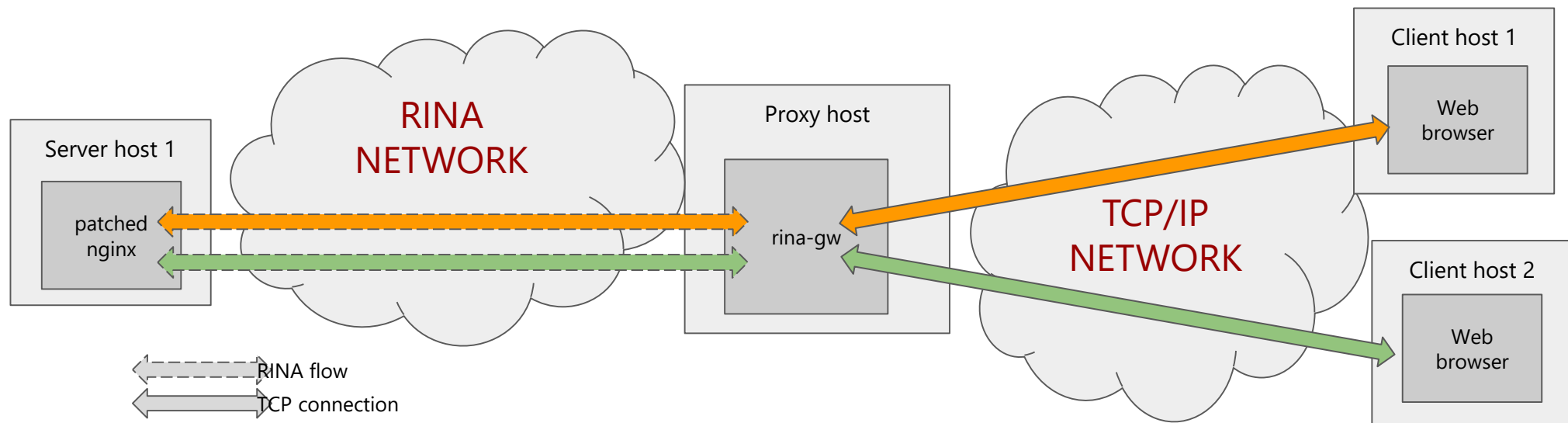


dropbear



- Dropbear source code patched to use the RINA API instead of sockets
- Dropbear server binary enhanced with two more command line arguments:
 - -D to specify the DIF to register to
 - -A to specify the application name to register
 - Ex. # `dropbear -R -A dropbear.1 -D n.DIF`
- Similar modifications to the client binary
 - -D to specify the DIF to use
 - server application name used as an hostname
 - Ex. \$ `dbclient -D n.DIF username@dropbear.1`

- nginx source code patched to use the RINA API instead of sockets
- The `listen` directive in the nginx configuration file accepts two additional options to specify the DIF to register to and the application name to register
 - Ex. `listen 80 rina_app1_name=nxnginx.1 rina_dif_name=n.DIF`
- No web client is currently available, rina-gw is necessary



RINA interposer



- Porting a socket application is not hard (guidelines [here](#))
 - However, it requires to study the source code, patch it, and recompile the application
- The [RINA interposer \(https://github.com/matt-williams/rina-interposer\)](https://github.com/matt-williams/rina-interposer) is an interesting alternative
 - No patch required
 - LD_PRELOAD variable set to replace the libc socket calls at load time
 - Socket operations are remapped over the RINA API at run-time
- The current prototype has room for improvement:
 - Complete socket API coverage
 - a flexible way to map hostnames, IPs and TCP ports to RINA application names